ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ МОЛДОВЫ
ДОКТОРСКАЯ ШКОЛА ФИЗИКО-МАТЕМАТИЧЕСКИХ, ИНФОРМАЦИОННЫХ И ТЕХНИЧЕСКИХ НАУК

На правах рукописи
C.Z.U: 519.21:004.421(043.3)

МАЛЮТИНА НАДЕЖДА

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В РАЗРАБОТКЕ КРИПТОГРАФИЧЕСКИХ И АЛГЕБРАИЧЕСКИХ АЛГОРИТМОВ

122.03 – МОДЕЛИРОВАНИЕ, МАТЕМАТИЧЕСКИЕ МЕТОДЫ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Докторская диссертация по информатике

Автор: Малютина Надежда
Научные руководители: Корлат Андрей, доктор физико-математических наук, профессор университар
Щербаков Виктор, доктор хабилитат физико-математических наук, профессор университар
Научные консультанты: Кожокару Светлана, доктор хабилитат информатики, член-корреспондент
Арнаутов Владимир, доктор хабилитат физико-математических наук, академик
Цицкиев Инга, доктор информатики, конференциар университар

КИШИНЕВ, 2023

UNIVERSITATEA DE STAT DIN MOLDOVA
ȘCOALA DOCTORALĂ ȘTIINȚE FIZICE, MATEMATICE, ALE INFORMAȚIEI ȘI INGINEREȘTI

Cu titlu de manuscris
C.Z.U.: 519.21:004.421(043.3)

MALIUTINA NADEJDA

UTILIZAREA TEHNOLOGIILOR INFORMAȚIONALE LA ELABORAREA ALGORITMILOR CRIPTOGRAFICI ȘI ALGEBRICI

122.03 – MODELARE, METODE MATEMATICE, PRODUSE PROGRAM

Teză de doctor în informatică

Autor: Maliutina Nadejda

Conducători ştiinţifici: Corlat Andrei, dr. în șt. fizico-matematice, prof. univ.
Șcerbacov Victor, dr. hab. în șt. fizico-matematice, prof. univ.

Comisia de îndrumare: Arnautov Vladimir, dr. hab. în șt. fizico-matematice, acad.
Cojocaru Svetlana, dr. hab. în informatică, m.cor., prof. cerc.
Țițchiev Inga, dr. în informatică, conf. univ.

CHIŞINĂU, 2023

СОДЕРЖАНИЕ

АННОТАЦИЯ

Малютина Надежда: "Использование информационных технологий в разработке криптографических и алгебраических алгоритмов"

Докторская диссертация по информатике, Кишинёв, 2023

Структура диссертации: диссертация состоит из введения, четырех глав, общих выводов и рекомендаций, списка литературы из 201 источника и 3 приложений. Диссертация содержит 145 страниц основного текста, 1 рисунок и 71 таблицу. Полученные результаты были опубликованы в 21 научной работе.

Ключевые слова: Алгоритм Марковского, квазигруппа, левая и правая квазигруппа, трансляция, открытый текст, зашифрованный текст, атака, ключ, шифрование, дешифрование.

Цель исследования: построение новых и усовершенствование уже построенных криптографических алгоритмов и их криптоанализ.

Задачи исследования: 1. Разработка эффективного криптографического алгоритма на основе алгоритма Марковского с использованием n-арных группоидов; 2. Написание программ, реализующих работу построенных алгоритмов; 3. Проведение атак на все построенные шифры; 4. Сравнительный анализ проведенных атак; 5. Нахождение текстов минимальной длины для всех исследованных типов атак.

Научная новизна и оригинальность работы: результаты работы новые и оригинальные. Они являются продолжением предыдущих исследований в этой области. Разработаны и обобщены алгоритмы, которые позволили улучшить работу классического алгоритма Марковского, изучены атаки на построенные шифры и показана степень стойкости этих шифров.

Полученный результат, который способствует решению важной научной проблемы: состоит в разработке новых обобщений классического алгоритма, которые способствуют увеличению стойкости построенного шифра к известным видам атак.

Теоретическая значимость работы: определяется получением новых алгоритмов и шифров, построенных с применением неассоциативных структур, таких как n-арные группоиды. Разработаны новые обобщения алгоритмов кодирования с использованием левых и правых квазигрупп, n-арных группоидов обратимых на одном фиксированном месте.

Прикладная ценность работы заключается в использовании полученных результатов в теории кодирования и криптоанализе.

Внедрение научных результатов: Полученные результаты могут быть использованы в научных исследованиях, связанных с кодированием данных, изучением эффективности представления информации, криптоанализе данных. Они также могут быть использованы при разработке факультативного курса для студентов университетов, связанного с изучением криптологии на абстрактных алгебраических структурах.

ADNOTARE

Maliutina Nadejda: "Utilizarea tehnologiilor informaționale la elaborarea algoritmilor criptografici și algebrici"

Teză de doctor în informatică, Chișinău, 2023

Structura tezei: teza constă din introducere, patru capitole, concluzii generale și recomandări, bibliografie din 201 titluri și 3 anexe. Teza conține 145 pagini de text de bază, o figură și 71 tabele. Rezultatele obținute sunt publicate în 21 lucrări științifice.

Cuvinte-cheie: algoritm Markovski, cvazigrup, cvazigrup de stânga și de dreapta, translație, text deschis, text cifrat, atac, cheie, criptare, decriptare.

Scopul lucrării: construirea noilor și îmbunătățirea algoritmilor criptografici deja construiți și a criptoanalizei acestora.

Obiectivele cercetării: 1. Dezvoltarea unui algoritm criptografic eficient bazat pe algoritmul Markovski folosind grupoizi n-ari; 2. Elaborarea programelor, care implementează lucrul algoritmi construiti; 3. Efectuarea atacurilor asupra tuturor cifrurilor construite; 4. Analiza comparativă a atacurilor comise; 5. Găsirea textelor de lungime minimă pentru toate tipurile de atacuri investigate.

Noutatea și originalitatea științifică: Rezultatele lucrării sunt noi și originale. Ele sunt o continuare a cercetărilor anterioare în acest domeniu. Au fost dezvoltați și generalizați algoritmi, care au permis îmbunătățirea activității algoritmului clasic Markovski, au fost studiate atacurile asupra cifrurilor construite și a fost arătat gradul de rezistență al acestor cifruri.

Rezultatul obținut care contribuie la soluționarea unei probleme ştiinţifice importante: constă în dezvoltarea noilor generalizări ale algoritmului clasic care cresc rezistența cifrului construit la tipuri cunoscute de atacuri.

Semnificația teoretică a lucrării: este determinată prin obținerea noilor algoritmi și cifrurilor construite, folosind structuri neasociative precum grupoizii n-ari. Sunt dezvoltate noi generalizări ale algoritmilor de codare folosind cvazigrupuri de stânga și de dreapta, grupoizi n-ari inversabili la un loc fixat.

Valoarea aplicativă: constă în utilizarea rezultatelor obținute în teoria codificării și criptoanaliză.

Implementarea rezultatelor ştiinţifice: rezultatele obținute pot fi utilizate în cercetările științifice legate de codificarea datelor, studierea eficienței prezentării informațiilor și criptoanaliza datelor. Ele pot fi utilizate și în proiectarea unui curs opțional pentru studenții universitari legat de studiul criptologiei pe structuri algebrice abstracte.

ANNOTATION

Malyutina Nadezhda: "The use of information technologies in the development of cryptographic and algebraic algorithms"

PhD Thesis in Computer Science, Chisinau, 2023

Thesis structure: the thesis consists of Introduction, four main chapters, general conclusions and recommendations, bibliography of 201 sources, and 3 annexes. The thesis contains 145 pages of the main text, one figure, and 71 tables. The obtained results were published in 21 scientific works.

Keywords: Markovski algorithm, quasigroup, left and right quasigroup, translation, plaintext, ciphertext, attack, key, encryption, decryption.

The purpose of the thesis: construction of new modifications and improvement of already developed cryptographic algorithms and their cryptanalysis.

The objectives of the work: 1. Development of an effective cryptographic algorithm based on the Markovski algorithm using n-ary groupoids; 2. Writing programs that perform the work of the constructed algorithms; 3. Carrying out attacks on all built ciphers; 4. Comparative analysis of the attacks carried out; 5. Finding texts of the minimum length for all investigated types of attacks.

The scientific novelty and originality: the main results of the work are new and original. They are a continuation of previous research in this area. Algorithms were developed and generalized that allowed us improving the work of the classical Markovski algorithm, the attacks on the constructed ciphers were studied, and the degree of resistance of these ciphers was shown.

The important scientific problem being solved in the research: it consists in the development of new generalizations of the classical algorithm, which contribute to an increase in the resistance of the constructed cipher to known types of attacks.

The theoretical significance of the thesis: is determined by obtaining new algorithms and ciphers built using non-associative structures such as n-ary groupoids. New generalizations of coding algorithms using left and right quasigroups, n-ary groupoids invertible in one fixed place are developed.

The applicative value of the thesis: it lies in the use of the obtained results in coding theory and cryptanalysis.

The implementation of the scientific results: the results obtained can be used in scientific research related to data coding, study of the efficiency of information presentation, and data cryptanalysis. They can also be used in the design of an elective course for university students related to the study of cryptology on abstract algebraic structures.

СПИСОК СОКРАЩЕНИЙ

DES	–	**D**ata **E**ncryption **S**tandard
AES	–	**A**dvanced **E**ncryption **S**tandard
RSA	–	**R**ivest **S**hamir **A**dleman
DSA	–	**D**igital **S**ignature **A**lgorithm
ECDSA	–	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm
MD	–	**M**essage **D**igest
SHA	–	**S**ecure **H**ash **A**lgorithm
GM	–	**G**oldwasser–**M**icali
IBE	–	**I**dentity-**B**ased **E**ncryption
RFC	–	**R**equest **F**or **C**omments
GSM	–	**G**roupe **S**pécial **M**obile
SPN (SP- network)	–	**S**ubstitution-**P**ermutation **N**etwork
PGP	–	**P**retty **G**ood **P**rivacy
SSL	–	**S**ecure **S**ocket **L**ayer
EDS (ЭЦП)	–	**E**lectronic **D**igital **S**ignature (**Э**лектронная **Ц**ифровая **П**одпись)
TLS	–	**T**ransport **L**ayer **S**ecurity
IDEA	–	**I**nternational **D**ata **E**ncryption **A**lgorythm
FEAL	–	**F**ast data **E**ncipherment **AL**gorithm
EDE	–	**E**ncrypt-**D**ecrypt-**E**ncrypt
ECC	–	**E**lliptic **C**urve **C**ryptography
DSS	–	**D**igital **S**ignature **S**tandard
IBM	–	**I**nternational **B**usiness **M**achines

ВВЕДЕНИЕ

С наступлением информационного века кодирование и шифры стали необходимыми для нормального функционирования общества, а постоянное совершенствование информационных технологий способствовало интенсивному развитию криптографических и алгебраических алгоритмов. Эта работа посвящена построению и изучению различных обобщений алгоритма Марковского, а также криптоанализу текста, построенного с помощью этих алгоритмов.

Актуальность и важность решаемой проблемы. С древних времен человечество использовало различные варианты кодирования информации, изобретались устройства, которые бы способствовали сохранению в тайне секретной информации. Толчком к развитию теории информации послужила работа нидерландского лингвиста Огюста Керкгоффса «Военная криптография» («La Cryptographie Militaire», 1883). В этой работе автор изложил свои взгляды на проектирование криптографических систем и привел свой знаменитый принцип Керкгоффса. Одним из главных трудов XX века в области криптоанализа выступила в 1920 году монография Уильяма Фредерика Фридмана “The Index of Coincidence and its Application in Cryptography” (1921). Фридман впервые ввел в обращение термины "cryptanalysis" (1920) и "cryptology" (1935) [1].

Криптография долгое время продолжала оставаться секретной наукой, так как ее основной задачей являлось сохранение в тайне государственных секретов. Клод Шеннон в своей статье “Communication Theory of Secrecy Systems (1949), опубликованной в журнале “Bell System Technical Journal”, сформулировал фундаментальные понятия теоретической криптографии и положил начало криптографии как отдельной и очень важной науки [2].

В 1968 Хорст Фейстель начал работать в IBM Watson Laboratory над проблемами безопасности данных. Он был одним из первых неправительственных исследователей, который занялся разработкой теории блочных шифров. Фейстель принимал участие в создании проекта Lucifer и первым предложил использовать SP-сети. Его исследования в области блочных шифров послужили основой для создания алгоритма шифрования **DES** (**D**ata **E**ncryption **S**tandard).

После появления стандарта шифрования DES в 1975 году, Уитфильд Диффи и Мартин Хеллман предложили новый метод криптографического преобразования информации – криптографию с открытым ключом или асимметричную криптографию [3]. В своей работе авторы описали абсолютно новый подход к распределению

криптографических ключей, ссылаясь на работы Ральфа Меркля, который параллельно с ними занимался задачей распределения ключей среди пользователей. Новый метод использовал модульную арифметику [4] и в нем появились два фундаментальных изменения: персонализация ключей и разделение ключей. В результате взлом ключа становится крайне сложной задачей. Алгоритм Диффи-Хеллмана считается основой современной криптографии и демонстрирует возможность получения криптографического метода, который не требует обмена ключами, хотя и использует открытую связь – передачу пары первых чисел, которые служат для определения ключа. Алгоритмы такого типа и задачи дискретного логарифмирования долго не получали должного внимания, а именно, до 1990-х годов.

Появление концепции ассимметричных криптографических систем и создание первых криптографических алгоритмов нового типа, которые были практически реализуемы, произвело революционный переворот в криптографии и повлекло быструю ее алгебраизацию, что повлекло за собой вовлечение в криптографическую теорию и практику все новых алгебраических объектов.

Почти все известные к тому времени конструкции кодов обнаружения и исправления ошибок, криптографические алгоритмы и системы шифрования использовали ассоциативные алгебраические структуры, такие как группы и поля [5, 6]. Анализ исследований показал, что можно использовать достаточно успешно такие неассоциативные структуры, как квазигруппы во многих областях теории кодирования, и особенно в криптологии.

Теория квазигрупп в настоящее время развивается в нескольких направлениях, и среди этих направлений наибольший интерес представляет применение теории квазигрупп в криптологии. Более того, коды и шифры, основанные на неассоциативных структурах, показывают лучшие возможности, чем коды и шифры, построенные на основе ассоциативных структур [7, 8].

Первыми профессиональными криптографами, которые занимались развитием теории квазигрупп, были: А.А. Альберт, А. Дриско, М.М. Глухов, Дж.Б. Россер, Э. Шёнхардт, Х. Дж. Мендельсон и Р. Шауфлер. Некоторые результаты, полученные в области применения квазигрупп в криптологии и теории кодирования, описаны в работах Дж. Денеса и А.Д. Кидвелла. [7, 9-11]. Многие результаты неассоциативной криптографии с открытым ключом можно найти у Аркадиуша Калки [12].

Важные результаты в применении теории квазигрупп в криптографии были получены М.Э. Тужилиным [13]; Ю.М. Мовсисяном [14]; А.В. Грибовым, П.А. Золотых и А.В. Михалевым [15]; Дж. Мейз, К. Моника и И. Розенталем [16]; В. Шпильрайном и А. Ушаковым [17]; Р.Э. Атани, Ш.Э. Атани и С. Мирзакучаки [18]; А. Крапежем [19, 20]; К. А. Мейером [21]; В.А. Артамоновым, С. Чакрабарти, В.Т. Марковым и С.К. Полом [22, 23].

Ч. Кошельны и Г.Л. Маллен представили криптосистему с открытым ключом, использующую обобщенные поточные шифры, основанные на квазигруппах [24]. Квазигруппы для безопасного кодирования предложили использовать Э. Оходкова и В. Снасель [25]; С. Марковски, Д. Глигороски, Б. Стойцевска и В. Бакева [26, 27]; С. Марковски, В. Димитрова, З. Трайческа, М. Петковска, М. Костадиноски и Д. Бухов [28].

Более полный обзор применения квазигрупп в криптологии можно найти в работе В.А. Щербакова [29].

С. Марковски и его соавторы представили поточный шифр с почти открытым ключом, основанным на квазигруппах в [26]. Алгоритм Марковского и его обобщения в настоящее время широко известны и часто используются в поточных шифрах на основе квазигрупп. Усовершенствования и исследования алгоритма Марковского интенсивно проводились В.А. Щербаковым в [29].

Важные результаты были получены А. Крапежем, В. Бакевой, В. Димитровой и А. Поповской-Митровики [30-32]. А. Крапеж и Д. Живкович предлагают использовать парастрофические преобразования квазигрупп и их модификации, которые весьма перспективны для применения и исследования [33]. Криптоанализ этих шифров изучался в диссертационной работе М. Войводы [34].

Некоторые обобщения и модификации алгоритма Марковского можно найти в работах В.А. Щербакова и А. Петреску [35-39]. Дальнейшее развитие алгоритма Марковского представляется в работах С. Марковского, Д. Глигороски, Л. Коцарева, С. Й. Кнапскога, М. Хассинена [40-42]; С. Чакрабарти, Сейбал К. Пал и С. Гангопадхья [43].

Важные сведения о криптоанализе некоторых поточных шифров можно найти в статье В.А. Щербакова и П. Ксорго [44].

Алгоритм Марковского имеет множество различных обобщений и может быть использован для построения аналогов схемы Эль-Гамаля. Аналог системы шифрования Эль-Гамаля на основе алгоритма Марковского приведен в работах В.А. Щербакова и Н.А. Молдовяна [45]; А.В. Грибова [46].

Интенсивное развитие современной криптологии связано с быстрым внедрением и совершенствованием персональных компьютеров и сетей. В этом направлении появилось много новых математических и криптографических задач, часть из которых до сих пор не решена.

Криптографические методы стали широко использоваться в электронной коммерции, телекоммуникациях и многих других средах. Эти методы используются не только для шифрования транзакций и контроля над производством криптовалют, но и обеспечивают безопасную работу банковских систем, пластиковых карт, банкоматов, беспроводных устройств и т.д.

Современная криптография занимается такими проблемами защиты информации, как конфиденциальность, целостность, аутентификация, управление ключами и невозможность отказа сторон от авторства. Создание надежных алгоритмов шифрования является ключевой задачей защиты информации. Поэтому любой построенный алгоритм необходимо подвергать тщательному анализу с целью выявления его слабых мест и возможности взлома.

Без криптографии не обойтись при защите данных, передаваемых по открытым каналам связи, а также там, где необходимо подтвердить целостность электронной информации или доказать ее авторство. Криптографические методы нашли широкое применение в практической информатике для решения многочисленных задач защиты информации.

Важной задачей современной криптографии является повышение стойкости и уменьшение размера блоков данных за счет модификации существующих криптосистем и построения новых с улучшенными характеристиками.

В диссертации поднимаются следующие вопросы:

Задача 1. Исследовать и построить алгоритмы на основе алгоритма Марковского с использованием квазигрупп и группоидов.

Задача 2. Провести криптоанализ шифров, построенных с помощью обобщенных алгоритмов.

Обзор основных работ по исследуемой проблематике будет сделан в первой главе диссертационной работы.

Задача 1 обсуждается в Главе 2, а задача 2 решается в Главах 3 и 4.

Цель и задачи диссертации. Целью научного исследования является построение новых и усовершенствование уже разработанных криптографических алгоритмов на основе

алгоритма Марковского, проведение их криптоанализа и написание программ, реализующих работу этих алгоритмов.

Для достижения этой цели были поставлены следующие задачи:

- Разработка эффективного криптографического алгоритма на основе алгоритма Марковского с использованием левой и правой бинарных квазигрупп и n-арных группоидов;
- Разработка программ, реализующих работу построенных алгоритмов;
- Проведение атак на все изученные и построенные шифры;
- Осуществление сравнительного анализа всех проведенных атак;
- Нахождение текстов минимальной длины для всех изученных типов атак.

Гипотеза исследования. Классический алгоритм Марковского может служить базой для построения новых обобщений на основе бинарных квазигрупп и обратимых на одном фиксированном месте группоидов. Построенные обобщенные алгоритмы будут иметь более высокую степень стойкости к известным видам атак. Криптоанализ шифров, построенных с использованием обобщенных алгоритмов, представляет собой интересную область исследований для криптоаналитиков.

Прикладные методы исследования. В данной работе применяется анализ научной литературы и практического опыта, проводится систематизация ранее полученных результатов по проблеме исследования. Сравниваются существующие подходы к решению поставленных задач и современные методы построения криптографических алгоритмов на основе неассоциативных структур и их свойств, и, в частности, методы неассоциативной алгебры, включая методы построения n-арных группоидов, а также классические методы криптоанализа. Исследование основано на использовании классического алгоритма Марковского и его обобщений.

Объектом исследования являются обобщенные алгоритмы Марковского, основанные на бинарных квазигруппах и n-арных группоидах.

Научная новизна и оригинальность. Все результаты работы новые и оригинальные. Они представляют собой продолжение предыдущих исследований в этой области. Были разработаны и обобщены алгоритмы, позволившие улучшить работу классического алгоритма Марковского, построены атаки на шифры с использованием обобщенных алгоритмов, а также показана степень стойкости этих шифров. Результаты, представленные в диссертации, представляют интерес для изучения криптологами.

Важная научная решаемая в исследовании задача состоит в разработке новых модификаций классического алгоритма, способствующих повышению стойкости построенного шифра к известным видам атак.

Теоретическая значимость состоит в получении новых улучшенных алгоритмов и шифров, с помощью применения неассоциативных структур, таких как n -арные группоиды, в информатике. Разработанные алгоритмы позволили с новой точки зрения подойти к проблемам, связанным с кодированием и криптоанализом.

Прикладное значение диссертации. Предложены новые модификации алгоритмов кодирования с использованием левых квазигрупп, правых квазигрупп и обратимых на одном месте n-арных группоидов. Разработанные методы позволили решить поставленные задачи и обозначили круг дальнейших вопросов, на которые еще предстоит ответить. Прикладное значение работы заключается в использовании полученных результатов в научных исследованиях, связанных с кодированием данных, изучением эффективности представления информации, криптоанализом данных. Они также могут быть использованы при разработке специализированных курсов для студентов, магистров и докторантов, связанных с изучением криптологии на абстрактных алгебраических структурах.

Основные научные результаты, представленные на защиту:

- разработаны обобщенные алгоритмы Марковского для левой и правой квазигрупп и программы для их реализации;
- проведены атаки на шифры, построенные с использованием левой и правой квазигрупп;
- подобраны тексты минимальной длины для успешного проведения каждой атаки;
- проведен сравнительный анализ атак, в том числе с помощью предельных переходов;
- построен обобщенный алгоритм Марковского для n -арного группоида, обратимого на одном фиксированном месте;
- построен обобщенный алгоритм Марковского для n -арного группоида, обратимого на одном фиксированном месте, с использованием трансляций любых степеней;
- проведен криптоанализ построенных обобщенных алгоритмов;
- выявлен минимальный текст для проведения успешных атак с использованием выбранных шифротекстов;

- установлены нижние границы количества символов для текстов, используемых в атаках для различных случаев;
- проведен анализ работы аналога схемы Эль-Гамаля на основе алгоритма Марковского и изучены его особенности.

Результаты представленной диссертации внедрены в работу научно-исследовательской лаборатории «Алгебра и ее приложения» Приднестровского государственного университета им. Т.Г. Шевченко (Тирасполь).

Апробация результатов. Полученные научные результаты были представлены и одобрены в рамках семинара «Алгебра и математическая логика», посвященного памяти профессора В. Белоусова в Институте Математики и Информатики Молдовы им. Владимира Андрунакиевича. Научные результаты, полученные автором в данной диссертации, были представлены на национальных и международных научных конференциях:

- International Conference on Mathematics, Informatics, and Information Technologies: dedicated to the illustrious scientist Valentin Belousov, MITI 2018, April 19-21, Bălţi, Republic of Moldova, 2018.
- Conference on Mathematical Foundations of Informatics, MFOI, Chisinau, Republic of Moldova, July 2-6, 2018 and Iași, Romania, July 3-6, 2019.
- "Tendinţe contemporane ale dezvoltării științei: viziuni ale tinerilor cercetători": Conferința Științifică a Doctoranzilor (cu participare internațională), Chișinău, June 15, 2018, June 10, 2019 (the plenary report) and June 15, 2020.
- LOOPS 2019 Conference, Budapest University of Technology and Economics, Hungary, July 7-July 13, 2019.
- The 5^{th} International Conference of Mathematical Society of the Republic of Moldova, dedicated to the 55^{th} anniversary of the foundation of Vladimir Andrunachievici Institute of Mathematics and Computer Science, IMCS-55, Chisinau, September 28 - October 1, 2019.
- International Symposium "Actual Problems of Mathematics and Informatics" dedicated to the 90th birthday of professor Ion Valuţă, Chişinău, Moldova, November 27-28, 2020.
- Conference "Contemporary Research and Evaluation Methodologies, Biological and Chemical Sciences, Physical and Mathematical Sciences, Economic Sciences", Chisinau, Republic of Moldova, Moldova State University, April 22-23, 2021.

- Conferinţa ştiinţifică studenţească cu participare internaţională, Chisinau, Tiraspol State University, April 28, 2021.
- “International Conference of Mathematics & Information Technologies: Research and Education”, MITRE, Republic of Moldova, Moldova State University, Chisinau, June 23–26, 2019 and July 1–3, 2021.
- Workshop on Intelligent Information Systems WIIS2021, Chisinau, Republic of Moldova October 14-15, 2021.

Публикации по теме диссертации и личный вклад. Результаты исследования опубликованы в 21 научной работе, в том числе 7 научных статей (2 статьи без соавторов), 8 статей в материалах научных конференций (6 статей без соавторов) и 6 тезисов на научных конференциях (2 тезиса без соавторов).

Непосредственно автором разработано математическое и алгоритмическое обеспечение шифрования и дешифрования текстов, построенных на основе обобщенных алгоритмов Марковского, а также проведен криптоанализ всех построенных шифров.

Структура и содержание диссертации.

Работа включает 217 страниц и состоит из четырех глав, содержащих теоретические и практические результаты, полученные при изучении и построении обобщений алгоритма Марковского, а также при их криптоанализе. Диссертация также содержит аннотацию на английском и румынском языках, введение, общие выводы и рекомендации, приложения и библиографию из 201 наименования.

Во введении формулируются актуальность и значимость темы исследования, определяется объект исследования, формулируются цель и задачи исследования, определяются методы исследования, раскрывается научная новизна, теоретическая и практическая значимость диссертации. Изучаемая научная проблема представлена с акцентом на важность прикладной ценности работы. Сформулированы основные положения к защите, приведены сведения об апробации и внедрении результатов. Представлен краткий анализ выступлений и публикаций по теме диссертации. В конце этого раздела приводится краткое изложение содержания работы.

Первая глава «Текущая ситуация в области использования информационных технологий при разработке криптографических и алгебраических алгоритмов» диссертации носит вводный характер и направлена на представление современной ситуации в области использования информационных технологий при разработке криптографических и алгебраических алгоритмов. В ней представлен обзор текущего

состояния наиболее важных областей современной криптографии для нашей работы. Описаны основные понятия криптологии, необходимые для дальнейшего изложения работы. Проведен анализ одной из наиболее часто используемых классификаций криптографических алгоритмов и отмечено, каким условиям должны удовлетворять современные алгоритмы шифрования.

Анализируются основные особенности и проблемы симметричного и асимметричного шифрования. Выявлены преимущества и недостатки, характерные для современных симметричных и асимметричных систем. Описаны наиболее популярные функции хэширования и цифровые подписи. Особое внимание уделено существующим на сегодняшний день криптоаналитическим методам. Сделан обзор применения неассоциативных алгебраических структур в криптологии, в котором основное внимание уделено алгоритму Марковского и его построенным на сегодняшний день обобщениям.

Во второй главе «Алгоритм Марковского и его новые обобщения» изучается работа алгоритма Марковского для бинарных квазигрупп и особенности работы алгоритма для левой и правой квазигрупп. Также описываются обобщения алгоритма Марковского для обратимых группоидов на любом фиксированном месте. Рассмотрены построение и работа обобщенного алгоритма Марковского с использованием трансляций различных степеней. Эти алгоритмы были построены совместно с В.А. Щербаковым. Модифицированные алгоритмы сравниваются и оцениваются по различным критериям.

В третьей главе «Криптоанализ некоторых потоковых шифров (бинарный случай)» проводится криптоанализ построенных в предыдущей главе шифров с использованием атаки М. Войводы. Для бинарных квазигрупп М. Войвода провел свою атаку и в диссертации показывается, как можно улучшить эти результаты. Кроме того, автор работы предлагает свои модифицированные атаки, которые показывают лучшие результаты, чем атаки М. Войводы. Рассматриваются атаки для левой и правой бинарных квазигрупп (модифицированные атаки). Для каждого случая определяется минимальное количество символов, необходимое для успешной атаки. Анализируются предельные соотношения количества символов, используемых в различных видах атак.

В четвертой главе «Криптоанализ некоторых потоковых шифров (n-арный случай)» продолжается криптоанализ для шифров, построенных в Главе 2, в случае использования i-обратимых n-арых группоидов. Атаки осуществляются на шифры, построенные с использованием обобщенного алгоритма Марковского. Было найдено необходимое количество символов для атаки с выбранным шифртекстом и установлена

минимальная граница для атаки с использованием выбранного открытого текста. Приведены различные примеры текстов минимальной длины для ряда случаев. Сделаны выводы по всем видам проведенных атак.

В разделе **«Общие выводы и рекомендации»** представлены общие выводы автора, основанные на результатах, полученных в диссертационной работе. Выводы сопровождаются авторскими рекомендациями о том, как эти результаты могут быть применены в различных областях науки, а также в потенциальных исследованиях.

Благодарности. Выражаю искреннюю благодарность моему научному руководителю *Виктору Алексеевичу Щербакову* за определение области и формулировку задач исследования, за знания, которые я получила за четыре года обучения в докторантуре, и за помощь, которую он мне оказал в написании публикаций и диссертации.

Я также хочу выразить благодарность моему научному руководителю *Андрею Николаевичу Корлату* за его терпение, руководство и поддержку в течение этих четырех лет обучения в аспирантуре.

Особая благодарность членам руководящего комитета за советы и ценные рекомендации.

1. ТЕКУЩЕЕ СОСТОЯНИЕ В ОБЛАСТИ ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В РАЗРАБОТКЕ КРИПТОГРАФИЧЕСКИХ И АЛГЕБРАИЧЕСКИХ АЛГОРИТМОВ

В этой главе анализируется и обобщается ситуация в исследуемой области.

Изобретение понятия асимметричных криптографических систем и создание первых криптографических алгоритмов этого типа, пригодных для практического использования, произвело революцию в криптографии и привело к быстрой алгебраизации криптографии, вовлечению в криптографическую теорию и практику все большего числа алгебраических объектов.

Важным аспектом безопасности информационных систем является оценка надежности используемых криптографических алгоритмов. Одни алгоритмы ориентированы на аппаратную реализацию, другие, показывают лучшую производительность при программной реализации.

Стойкость современных шифров, помимо самого алгоритма шифрования, определяется длиной используемого ключа шифрования. Современная криптография исходит из того, что секретность шифра обеспечивается исключительно ключом шифрования, так как сам алгоритм рано или поздно будет раскрыт.

В этой главе будут выделены наиболее характерные черты современных криптографических систем, а также рассмотрены основные проблемы, связанные с определением криптостойкости современных систем защиты информации и подходы к их решению. Будет уделено внимание перспективам применения теории квазигрупп в криптологии и различным способам обобщения алгоритма Марковского и схемы Эль-Гамаля.

1.1. Обзор текущего состояния современной криптографии

Криптология – это область знаний, изучающая криптографию и методы ее раскрытия – криптоанализ [47]. Криптография занимается разработкой криптосистем, а криптоанализ занимается взломом этих криптосистем.

Построение современной криптологии как науки базируется на совокупности фундаментальных понятий и фактов математики, физики, теории информации и вычислений, которые сложны для всестороннего и глубокого понимания даже профессионалами [4, 48]. Однако все эти факторы не помешали тому, что многие

теоретические достижения криптологии широко используются в нашей информационно насыщенной жизни.

Перед современной криптографией стоят три основные проблемы [49]:

- ✓ обеспечение конфиденциальности (только владельцы ключей имеют доступ к информации);
- ✓ обеспечение анонимности;
- ✓ обеспечение аутентификации информации и источника сообщения.

Эти задачи обязаны своей постановкой массовому использованию электронных способов обработки и передачи информации (банковское дело, электронная коммерция, каналы межличностного общения и др.).

Любое криптографическое преобразование состоит из двух этапов: прямого преобразования, которое называется шифрованием, и обратного, которое называется дешифрованием. Исходное сообщение называется открытым текстом, а зашифрованное сообщение – зашифрованным текстом. Длина открытых и зашифрованных сообщений не меняется (за исключением методов шифрования с использованием электронной цифровой подписи).

Криптографический алгоритм – это математическая функция, которая используется для шифрования и дешифрования информации. До появления компьютеров в основе криптографии лежали алгоритмы, построенные с помощью операций замены одних символов другими – это алгоритмы подстановки, или перестановки символов местами – это алгоритмы перестановки. Современные криптосистемы используют оба типа алгоритмов. Более того, существуют достаточно сложные перестановочные шифры, но современные компьютеры с ними быстро справляются. Использование таких кодов требует большого объема памяти. Основой современной криптографии являются модульная арифметика и теория чисел, в частности, ее раздел, посвященный простым числам.

Если защита, обеспечиваемая алгоритмом, основана на сохранении самого алгоритма в секрете, то это ограниченный алгоритм. Такие алгоритмы не позволяют осуществлять эффективный контроль или стандартизацию. Пользователи должны использовать собственный уникальный алгоритм и не могут использовать открытые программно-аппаратные продукты. Несмотря на эти недостатки, ограниченные алгоритмы чрезвычайно популярны в приложениях с низким уровнем безопасности.

Современная криптография решает многие возникающие проблемы с помощью ключа. Ключ – это секретный параметр, который контролирует ход преобразования.

Одним из требований обеспечения стойкости криптосистемы является огромное количество возможных ключей, не позволяющее провести исчерпывающий перебор. Однако само по себе большое количество ключей не обеспечивает стойкости криптосистемы. Известны достаточно надежные криптосистемы с малым пространством ключей [4].

Поскольку ключи обычно легче изменять и передавать, имеет смысл хранить их в секрете, чтобы обеспечить безопасность шифрования. Этот принцип был сформулирован в конце IXX века Огюстом Керкгоффсом. Принцип Керкгоффса определил ключ как фундаментальный элемент безопасности любой криптографической системы.

Криптосистемы, использующие одно и то же значение ключа для шифрования и дешифрования, называются симметричными. Их также называют криптосистемами с секретным ключом, поскольку значение ключа должно быть известно только отправителю и получателю сообщений. Симметричное шифрование идеально подходит для случая шифрования информации «для себя». Это может быть, как архивное шифрование выбранных файлов, так и автоматическое шифрование целых логических или физических дисков.

Другой класс современных криптографических алгоритмов использует разные ключи для шифрования и дешифрования. Поэтому асимметричные криптосистемы также называют криптосистемами с открытым ключом [50, 51]. Для современной криптографии характерно использование общедоступных алгоритмов шифрования и комбинированных преобразований, при которых исходный текст шифруется двумя или более различными способами.

Компьютерные шифры были и остаются уязвимыми, поскольку неавторизованный получатель может перехватить ключ и, зная алгоритм шифрования, расшифровать сообщение. Этот серьезный недостаток криптографических систем известен как проблема распределения ключей. Для обеспечения безопасности кода ключи шифрования должны быть защищены более надежно, чем алгоритм. Протокол распределения ключей – это согласованная последовательность действий пользователя для создания защищенного канала связи, которая заключается в генерации и обмене сеансовыми ключами и аутентификации сообщений.

Все протоколы распределения ключей делятся на три пересекающиеся категории: протоколы, основанные на асимметричной криптографии; протоколы на основе симметричной криптографии; и протоколы, использующие центр сертификации.

На сегодняшний день существуют проверенные алгоритмы шифрования, которые при использовании ключа достаточной длины и при правильной реализации относятся к категории криптостойких алгоритмов.

Одной из наиболее распространенных классификаций криптографических алгоритмов является следующая:

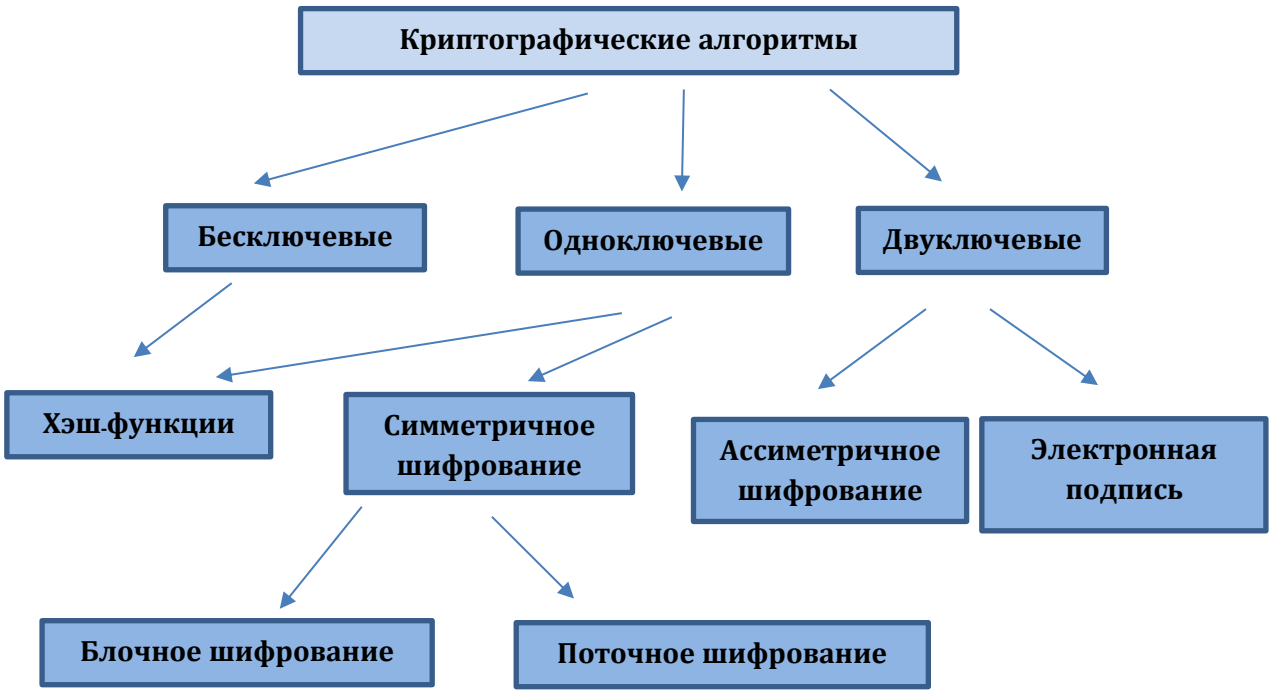


Рис. 1.1. Классификация криптографических алгоритмов

Примеры наиболее распространенных типов алгоритмов:

- *симметричные алгоритмы*:
  - блочные алгоритмы: $DES, DESX, RC5$, Blowfish, $AES, 3DES$,Twofish, Магма, $IDEA$, Камелия и другие;
  - поточные алгоритмы: $RC4, A5/1, A5/2$,Sober-128, Pike, Wake, SEAL и др.;
- *ассиметричные алгоритмы*: RSA, DSA, ElGamal, $ECDSA$ и др.;
- *Хэш-функции*: $MD4, MD5, MD6, SHA\text{-}2, SHA\text{-}3, SWIFFT$ и др.

Ш. Гольдвассер (S. **G**oldwasser) и С. Микали (S. **M**icali) представили концепцию вероятностного шифрования в 1982 году и создали первую схему вероятностного шифрования с открытым ключом – криптосистему **GM** [52]. Криптосистема называется вероятностной, если в процессе шифрования используются случайные параметры. При использовании таких криптосистем один и тот же исходный текст, зашифрованный одним и тем же ключом, в разное время может привести к совершенно разным шифротекстам. Для характеристики надежности криптосистемы Гольдвассер и Микали ввели широко

используемое понятие семантической стойкости. Существует вероятностная схема шифрования, которая работает быстрее, чем схема шифрования с открытым ключом RSA. Использование вероятностной криптосистемы **BG** Блюма-Гольдвассера (**B**lum-**G**oldwasser) эффективно для более длинных зашифрованных текстов, в которых RSA требует нескольких отдельных шифровок [53].

Бесключевая криптография, предложенная Б. Альперном и Ф. Шнейдером, скрывает только происхождение, а не содержание сообщений [54, 55]. Сервис кошелька Curv, который помогает защитить цифровые активы с помощью криптографии без ключа, все чаще используется во всем мире различными биржами, внебиржевыми офисами, кредиторами и брокерами. Curv обеспечивает математически безопасный способ подтверждения и подписи транзакций в блокчейне. Сервис Curv может поддерживать транзакции с мультиподписью в Bitcoin SV (форк криптовалюты Bitcoin Cash, появившейся в 2018 году), поскольку Curv предлагает пользователям возможность безопасного управления и торговли всеми типами цифровых активов в блокчейнах с использованием **ECDSA** (**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm) −алгоритм цифровой подписи на основе эллиптических кривых, аналогичный по структуре DSA, но определенный не над конечным числовым полем, а в группе точек эллиптической кривой и **EdDSA** (**Ed**wards-curve **D**igital **S**ignature **A**lgorithm) − алгоритм цифровой подписи на основе эллиптической кривой Эдвардса с использованием варианта схемы Шнорра [56].

В криптосистеме шифрования на основе идентичности (IBE) А. Шамира нет необходимости в распределении ключей, но требуется некий центр, которому необходимо доверить генерацию секретных ключей. Ч. Беннетт и Ж. Брассар разработали теорию квантовой криптографии, которая предлагает совершенно иную основу для современной криптологии и больше полагается на квантовую физику в ее заявлениях о секретности, чем на математику или теорию вычислительной сложности [57-59]. Ч. Беннетт вместе с Дж. Смолиным создали первый генератор квантовых ключей. После этого началось бурное развитие квантовой криптографии с использованием оптоволокна.

В 2019 году российские ученые из Центра научных исследований и перспективных разработок компании «Инфотекс» и Центра квантовых технологий МГУ имени М.В. Ломоносова успешно провели публичные испытания первого российского квантового телефона ViPNet **QSS** (**Q**uantum **S**ecurity **S**ystem). Конфиденциальность переговоров по «квантовому» телефону основана на стойком симметричном шифровании сетевого трафика между абонентами с использованием протокола распределения квантового ключа.

Российский квантовый телефон, над которым работали более трех лет, не подвержен известным атакам с использованием квантовых компьютеров, которые в ближайшем будущем смогут достичь производительности, достаточной для «взлома» многих используемых сегодня криптографических механизмов.

Таким образом, современные алгоритмы шифрования должны удовлетворять ряду условий:

- должны быть адаптированы к новейшей программно-аппаратной базе (изменение длины ключа не должно приводить к качественному ухудшению работы алгоритма);
- объем ключа должен соответствовать современным методам и средствам расшифровки зашифрованных сообщений (незначительное изменение ключа должно привести к существенным изменениям в зашифрованном сообщении);
- между ключами, используемыми в процессе шифрования, не должно быть простых зависимостей; ключ должен обеспечивать надежную защиту информации;
- операции шифрования и дешифрования должны быть максимально простыми, чтобы соответствовать современным требованиям и соответствовать скоростным характеристикам;
- не должны допускать появления все большего числа ошибок;
- должны свести к минимуму объем сообщений во время операций шифрования;
- зашифрованное сообщение должно быть раскрыто только при наличии ключа;
- количество операций, необходимых для определения ключа шифрования, должно быть не меньше общего количества возможных ключей;
- количество операций, необходимых для взлома информации путем полного перебора всех возможных ключей, должно иметь строгую нижнюю границу и выходить за пределы возможностей современных компьютеров;
- знание алгоритма шифрования не должно влиять на надежность защиты.

На практике криптографические алгоритмы в зависимости от области применения имеют несколько видов реализации: программную, аппаратную и программно-аппаратную.

Аппаратная реализация имеет лучшие скоростные характеристики, чем программные алгоритмы шифрования. Аппаратные средства защиты информации в большей степени защищены от побочного электромагнитного излучения, возникающего при эксплуатации оборудования, и от прямого физического воздействия на устройства, на

которых осуществляются операции шифрования и хранения ключевой информации. Аппаратное обеспечение более удобно для пользователя, поскольку оно позволяет прозрачно выполнять операции шифрования и дешифрования и его легко установить. Они обычно используются для защиты телефонных разговоров, отправки факсимильных сообщений и других видов передачи информации, где нельзя использовать программное обеспечение.

Программная реализация имеет такие преимущества, как гибкость и переносимость. Программа, написанная для одной операционной системы, может быть модифицирована для любой другой системы. Обновление программного обеспечения может быть выполнено с меньшими затратами времени и денег. Следует иметь в виду, что многие современные достижения в области криптографических протоколов недоступны для реализации в виде аппаратных средств.

К недостаткам криптографической защиты программного обеспечения можно отнести слабую физическую безопасность. Программная реализация не может соответствовать некоторым характеристикам, необходимым для надежного использования алгоритмов шифрования. Например, генерация ключевой информации не должна выполняться программными датчиками случайных чисел; для этого необходимо использовать специальные аппаратные устройства.

Программно-аппаратная реализация позволяет пользователям устранить некоторые недостатки информационной безопасности программного обеспечения, сохранив при этом их преимущества. Основными функциями, возлагаемыми на аппаратную часть программно-аппаратного комплекса криптографической защиты информации, обычно являются формирование ключевой информации и хранение ключевой информации в устройствах, защищенных от несанкционированного доступа. С помощью этих методов можно аутентифицировать пользователей с помощью паролей.

1.2. Основные особенности и проблемы симметричного и асимметричного шифрования

Симметричное шифрование было единственным методом шифрования до изобретения шифрования с открытым ключом. Многие страны приняли свои собственные национальные стандарты шифрования. В 1976 году в США был утвержден стандарт **DES** (**D**ata **E**ncryption **S**tandard). Этот стандарт использовался до тех пор, пока в 2001 году не был принят новый стандарт симметричного шифрования **AES** (**A**dvanced **E**ncryption **S**tandard)

на основе алгоритма Rijndael с длиной ключа 128, 192 и 256 бит. Алгоритм AES заменил предыдущий алгоритм DES, который рекомендовано использовать только в режиме Triple DES, в основном для защиты финансовой информации. Методы 2DES и 3DES на основе DES отличаются увеличенной длиной ключа (2DES – 112 бит, 3DES – 168 бит), в связи с чем возросла их криптостойкость.

Алгоритмы симметричного шифрования характеризуются следующими свойствами:

1) использование одного алгоритма для шифрования и дешифрования;

2) использование одного ключа, который держится в секрете.

Основные известные на сегодняшний день методы анализа симметричных систем можно найти в [60]. Подробное описание многих современных алгоритмов симметричного шифрования можно найти в монографии «Современные алгоритмы шифрования и методы их анализа» [61]. Книга С. Панасенко "Алгоритмы шифрования" – это достаточно подробное руководство по системам симметричного шифрования, в котором описано более 50 алгоритмов шифрования [62].

Современные алгоритмы симметричного шифрования делятся на блочные и поточные. Для блочных алгоритмов шифрование выполняется небольшими порциями – блоками (кратными 32 битам). В блочном шифре из двух одинаковых блоков открытого текста получаются одинаковые блоки шифротекста, что, безусловно, является одним из недостатков таких алгоритмов. Чтобы избежать этого, используются поточные шифры, в которых преобразование шифрования одного символа открытого текста изменяется от одного элемента к другому.

Поточный шифр – это, по сути, симметричный шифр, в котором каждый символ открытого текста преобразуется в символ зашифрованного текста. Преобразование зависит не только от используемого ключа, но и от положения символа в потоке открытого текста. Такие шифры обычно шифруют информацию в режиме реального времени и используют для шифрования специально сгенерированную псевдослучайную последовательность. Таким образом, потоковые шифры подходят для шифрования непрерывных потоков данных (например, голоса или видео). Посимвольное шифрование не вносит задержек в криптосистему, поэтому важнейшим преимуществом потоковых шифров является высокая скорость шифрования, эквивалентная скорости входящего ввода.

Примером потокового шифра является хорошо известный шифр A5/1, который используется для шифрования сообщений **GSM** (**G**roupe **S**pecial **M**obile). Из-за того, что спецслужбы всегда интересовались возможностью прослушивания в своих целях, в

алгоритм внесены изменения, позволяющие сломать его в разумные сроки. Алгоритм A5/3, разработанный в 2001 году, должен заменить A5/1 в мобильных системах третьего поколения. Его также называют алгоритмом KASUMI (он использует 64-битный размер блока и 128-битный ключ в 8-раундовой схеме Фейстеля). В настоящее время доступно большое количество различных потоковых шифров [63].

Блочные и поточные шифры реализованы по-разному. Потоковые шифры не очень подходят для программной реализации, но больше подходят для аппаратной реализации. В них длина шифротекста намного больше длины секретного ключа, а последовательность ключей псевдослучайна и имеет определенный период. Основная задача потоковых шифров – сгенерировать некоторую последовательность для шифрования. Очевидно, что если последовательность гамма-битов не имеет периода и выбрана случайно, то взломать шифр невозможно. Но кажется проблематичным иметь ключ, равный по размеру шифруемым данным.

Практически все каналы передачи данных для систем потокового шифрования подвержены помехам. Поэтому для предотвращения потери информации решается проблема синхронизации шифрования и дешифрования текста. По способу решения этой задачи шифровальные системы делятся на синхронные и самосинхронизирующиеся [64].

Синхронные поточные шифры – это шифры, в которых поток ключей генерируется независимо от открытого и зашифрованного текста. Основным свойством этих шифров является нераспространение ошибок.

В самосинхронизирующихся поточных шифрах символы ключевой гаммы зависят от исходного секретного ключа шифра и от конечного числа последних символов зашифрованного текста. Основная идея заключается в том, что внутреннее состояние генератора ключевого потока является функцией предыдущих s бит зашифрованного текста. Поэтому генератор ключевого потока на принимающей стороне, получив s бит, автоматически синхронизируется с генератором шифрования. Недостатком этих потоковых шифров является распространение ошибок.

На данный момент существует два основных метода построения алгоритмов симметричного шифрования – схема Фейстеля и сеть на основе подстановок и перестановок (**SPN** — **S**ubstitution-**P**ermutation **N**etwork). Алгоритмы DES и RC5 построены по схеме Фейстеля. Наиболее ярким представителем использования сети SPN является стандарт AES.

Большинство современных алгоритмов шифрования основано на сети Фейстеля, благодаря множеству преимуществ такой структуры. В отличие от сети Фейстеля, SP-сети обрабатывают весь зашифрованный блок за один раунд. Обработка данных сводится в основном к заменам и перестановкам. SP-сети встречаются гораздо реже, чем сети Фейстеля. Примером SP-сети является алгоритм Serpent и по мнению его создателей, алгоритм можно сломать только в том случае, если будет создана мощная новая математическая теория [65].

Две проблемы, связанные с практическим использованием симметричных криптосистем – хранение и обмен ключевой информацией, стали важными стимулами для разработки принципиально нового класса методов шифрования: криптографии с открытым ключом или асимметричной криптографии.

Концепция асимметричной криптографии была впервые предложена в 1976 году У. Диффи и М. Хеллманом и опубликована в том же году в «New Directions in Cryptography» [3]. Р. Меркль также входит в число основоположников асимметричной криптографии, который независимо от Диффи и Хеллмана пришел к тем же построениям, но опубликовал свои результаты позже.

С 1976 года многие криптографические алгоритмы были созданы с использованием концепции открытых ключей. Многие из них не являются стойкими, а многие стойкие алгоритмы очень часто не подходят для практической реализации, так как используют слишком большой ключ, либо размер получаемого с их помощью шифротекста значительно превышает объем открытого текста. И только очень малая часть этих алгоритмов одновременно надежна и пригодна для практического использования. Как правило, эти алгоритмы основаны на решении двух сложных математических задач, таких как задача дискретного логарифмирования и задача факторизации больших чисел, которые будут легко решаться на больших квантовых компьютерах с использованием алгоритма Шора [66]. Это уже задачи постквантовой криптографии.

Первым практическим применением системы шифрования с открытым ключом, широко используемой и по сей день, является алгоритм, введенный М. Гарднером и известный как **RSA** — буквенная аббревиатура от имен Ривест, Шамир и Адлеман (**R**ivest, **S**hamir, **A**dleman). Его надежность практически гарантирована, потому что процесс расшифровки невероятно сложен. Система RSA используется для защиты программного обеспечения и в схемах цифровой подписи.

Летом 1991 года Филип Циммерманн, американский физик и защитник конфиденциальности, предложил бесплатную систему шифрования под названием **PGP** (**P**retty **G**ood **P**rivacy). PGP использует классическое симметричное шифрование, что дает ему большую скорость на домашних компьютерах, но шифрует ключи с помощью асимметричного алгоритма RSA. В 2010 году группе ученых из Швейцарии, Японии, Франции, Нидерландов, Германии и США удалось разложить 768-битный 232-значный ключ RSA [67].

Существует три алгоритма, обеспечивающих достаточные возможности как для шифрования текста, так и для его цифровой подписи: RSA, Эль-Гамаля и Рабина. Однако все эти алгоритмы работают довольно медленно, шифруя и расшифровывая данные намного медленнее, чем симметричные алгоритмы. В результате они часто непригодны для шифрования больших объемов данных, а используются для отправки короткой зашифрованной информации. Например, для передачи секретного ключа шифрования для симметричных криптосистем.

Алгоритмы асимметричного шифрования характеризуются следующими свойствами:

1) нет необходимости использовать один и тот же алгоритм для шифрования и дешифрования данных;

2) использование двух ключей, один из которых открытый, а другой секретный.

Для анализа асимметричных криптосистем на сегодняшний день существует достаточно большое разнообразие методов. Среди них наиболее известны: метод Гельфонда, «Giant step-Baby step», метод встречи на случайном дереве, метод базы разложения, метод решета числового поля, метод Ферма, метод непрерывных дробей, метод квадратичного решета и др. [60]. Однако если при анализе симметричных криптосистем различные методы используют разные приемы, то при анализе асимметричных криптосистем все методы сводятся к решению двух задач разными путями: задача дискретного логарифмирования и задача факторизации больших чисел [68].

«Легковесная» криптография – раздел криптографии, целью которого является разработка алгоритмов для использования в устройствах, не способных обеспечить большинство существующих шифров достаточными ресурсами (память, источник питания, размер) для функционирования. Большинство современных алгоритмов шифрования предназначены для использования в составе программных комплексов без учета оптимизации на аппаратном уровне. Этот факт делает невозможным использование

большинства существующих криптографических алгоритмов в устройствах с ограниченной вычислительной мощностью, малым объемом и энергопотреблением.

«Легковесная» криптография становится особенно актуальной в свете развития идеи «Интернета вещей», представляющего собой беспроводную самонастраивающуюся сеть между объектами разных классов. Часто разработчики легковесных алгоритмов вынуждены выбирать между тремя требованиями к алгоритму: безопасностью, стоимостью и производительностью. На практике несложно оптимизировать любые две из трех целей разработки, но очень сложно оптимизировать все три цели разработки одновременно. В связи с этим существует множество реализаций облегченных криптографических алгоритмов: как программных, так и аппаратных. Они имеют разные, а иногда и противоположные характеристики.

Классические криптографические шифры имеют высокие скорости шифрования данных, но их нельзя использовать в системах с ограниченными ресурсами. Среди малоресурсных шифров есть аппаратно- и программно-ориентированные алгоритмы, использование которых также различается в зависимости от стоящей перед системой задачи [69].

Если сравнить асимметричные системы с симметричными, то получим:

Преимущества: не требуется предварительной передачи секретного ключа; секретный ключ дешифрования знает только одна сторона; в больших сетях количество ключей значительно меньше, чем в симметричной системе.

Недостатки: труднее вносить изменения в алгоритм; имеет более длинные ключи; шифрование-дешифрование на 2-3 порядка медленнее, чем в симметричном алгоритме с тем же текстом; требуется гораздо больше вычислительных ресурсов, поэтому на практике асимметричные системы используются в сочетании с другими.

Современная криптография позволяет создать шифр, неуязвимый для любой существующей формы криптоанализа. Даже самые быстрые компьютеры не могут пробить все возможные варианты алгоритмов шифрования, таких как RSA или DES, и систем, таких как PGP [4].

Глубокое понимание механизмов кодирования и шифрования, может быть чрезвычайно полезным, когда речь идет о защите всего, что представляет для нас ценность. Различные системы шифрования с открытым ключом или комбинации открытого и закрытого ключей, такие как PGP, обеспечивают высокий уровень конфиденциальности

при передаче информации. Однако безопасность сложных систем связи, таких как Интернет, заключается не только в конфиденциальности [4].

1.3. Хэш-функции и цифровая подпись

В 1989 году Р. Меркль и И. Дамгард независимо друг от друга предложили итеративный принцип построения криптографических хэш-функций. Этот принцип позволяет свести задачу построения хэш-функции на множестве сообщений различной длины к задаче построения отображения, действующего на множестве фиксированной конечной длины. Такие функции широко используются в сфере защиты компьютерной информации. Например, хэш-функции MD5, SHA-1, семейство хэш-функций SHA-2.

Преобразование, выполняемое хэш-функцией, называется хэшированием. Исходные данные называются входным массивом. Результат преобразования – это «хэш». Случай, когда хэш-функция преобразует более одного массива входных данных в одни и те же сводки, называется «коллизией». Вероятность коллизии используется для оценки качества хэш-функций.

Хэш-функция считается хорошей, если она удовлетворяет двум основным условиям: быстрота вычислений и минимум коллизий. При этом первое свойство зависит в основном от параметров компьютера, а второе от значений данных и алгоритма хэширования.

Важным свойством является также однонаправленность, согласно которой хэш-функция предполагает простоту своего прямого вычисления и сложность своего обратного вычисления (вычисление X из известного $H(X)$). Однонаправленность – важнейшее свойство многих криптографических алгоритмов.

Семейство MD состоит из хэш-функций MD2, MD4, MD5 и MD6. Оно было принято в качестве интернет-стандарта RFC 1321. Это 128-битная хэш-функция. Широко используется в мире программного обеспечения для обеспечения целостности передаваемого файла.

Семейство SHA состоит из четырех алгоритмов SHA: SHA-0 – SHA-3. Из них наиболее широко используется SHA-1, например, в протоколе **S**ecure **S**ocket **L**ayer (**SSL**). В 2012 году NIST выбрал алгоритм Keccak в качестве нового стандарта SHA-3. Его основные преимущества – эффективная работа и хорошая устойчивость к атакам.

Чтобы хэш-функция считалась криптографически стойкой, она должна удовлетворять трем основным требованиям, на которых основано большинство применений хэш-функций в криптографии:

- *необратимость*: для данного значения хэш-функции m должно быть вычислительно невозможно найти блок данных X, для которого $H\ (X)\ =\ m$;
- *устойчивость к коллизиям первого рода*: для данного сообщения M должно быть вычислительно невозможно найти другое сообщение N для которого $H\ (N)\ =\ H\ (M)$;
- *устойчивость к коллизиям второго рода*: должно быть вычислительно невыполнимо найти пару сообщений (M, M'), которые имеют одинаковый хэш.

Криптографические хэш-функции должны иметь лавинный эффект: малейшее изменение аргумента сильно меняет значение функции. В частности, хэш-значение не должно пропускать информацию даже об отдельных битах аргумента. Это требование является гарантией криптографической стойкости алгоритмов хэширования, которые хэшируют пароль пользователя для получения ключа [70].

Хэширование часто используется в алгоритмах цифровой подписи, где шифруется не само сообщение, а его хэш-код, что сокращает время вычислений, а также повышает криптостойкость. Также в большинстве случаев вместо паролей хранятся значения их хэш-кодов.

Электронная **ц**ифровая **п**одпись (**ЭЦП**) является элементом криптографического преобразования информации. С внедрением электронного документооборота проблема установления подлинности и авторства безбумажной документации стала особенно актуальной. При всех преимуществах современных криптосистем они не позволяют проводить аутентификацию данных. Поэтому средства аутентификации должны использоваться совместно с криптографическими алгоритмами.

Существует несколько схем построения ЭЦП:

- на основе алгоритмов симметричного шифрования;
- на основе алгоритмов асимметричного шифрования;
- на основе асимметричных алгоритмов шифрования и хэш-функций – это самая распространенная схема.

Комбинируя известные алгоритмы шифрования и цифровые подписи, можно создавать сообщения, которые будут зашифрованы и подписаны.

Асимметричные схемы ЭЦП относятся к криптосистемам с открытым ключом. Однако в отличие от алгоритмов асимметричного шифрования, в которых шифрование выполняется с использованием открытого ключа получателя, а дешифрование – с помощью закрытого ключа получателя, в схемах цифровой подписи подпись выполняется

с использованием закрытого ключа отправителя, а проверка – с использованием открытого ключа отправителя. Асимметричная криптография основана на использовании ключей получателя, а асимметричная схема ЭЦП основана на использовании ключей отправителя. При этом криптостойкость ЭЦП к подделке определяется теми же факторами: для того чтобы использование ЭЦП имело смысл, необходимо, чтобы вычисление легитимной подписи без знания закрытого ключа представляло собой вычислительно сложный процесс. Для ускорения процесса генерации подписи, то есть уменьшения количества вычислительных операций, обычно принято подписывать не само сообщение, а его образ, который получается путем вычисления хэш-функции сообщения.

Диффи и Хеллман предложили оригинальный способ использования шифрования с открытым ключом для аутентификации сообщения. Они заметили, что RSA и другие подобные алгоритмы обладают интересной симметрией. Закрытый ключ также можно использовать для шифрования сообщения, а открытый ключ – для его расшифровки. Такой подход не повышает безопасность – ведь открытый ключ доступен каждому, но получатель может убедиться, что сообщение пришло от конкретного отправителя, владельца закрытого ключа. Для проверки подлинности отправителя сообщения достаточно добавить к обычному шифрованию дополнительные шаги [71].

Существует криптографическая система, которая защищает передачу конфиденциальной информации, известная как **TLS** (**T**ransport **L**ayer **S**ecurity). Она была разработана в 1994 году корпорацией программного обеспечения для Интернета Netscape и через два года была принята в качестве глобального стандарта [4]. TLS и ее предшественник SSL используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентификации сообщений для сохранения целостности сообщений.

Следующие алгоритмы доступны в версии протокола 2021 года:

- для обмена ключами и проверки их подлинности используются комбинации алгоритмов: RSA (асимметричный шифр), Diffie-Hellman (защищенный обмен ключами), DSA (алгоритм цифровой подписи), ECDSA;
- для симметричного шифрования: RC4, IDEA, Triple DES, SEED, Camellia или AES;
- для хэш-функций: MD5, SHA, SHA-256/384.

Одно из применений TLS-соединений – подключение узлов к виртуальной частной сети – **VPN**. В дополнение к TLS также можно использовать набор протоколов IP Security.

Каждый из этих подходов к реализации VPN имеет свои преимущества и недостатки [72]. TLS широко используется в Интернет-приложениях, таких как веб-браузеры, электронная почта, обмен мгновенными сообщениями и IP-телефония (VoIP).

Сегодня говорят о создании такой системы, которая обеспечивала бы жесткую привязку процесса формирования и использования секретного закрытого ключа к любому биометрическому параметру человека (обработке папиллярного узора отпечатка пальца или радужной оболочки глаза). Создание ЭЦП на основе нового способа использования биометрических данных должно решить следующие вопросы: отражение биометрического параметра в электронном документе и возможность его распознавания; разработка секретного ключа на основе изображения биометрического параметра; создание и ведение базы данных биометрических параметров. Такие комплексы дополнят уже существующие системы защиты информации и обеспечат комплексную систему защиты электронных документов.

1.4. Основные проблемы и особенности криптоаналитических методов

Основная цель криптографии – сохранить открытый текст и ключ в секрете, а задача криптоанализа – восстановить открытый текст без доступа к ключу, если это возможно. Попытка криптоаналитика взломать текст называется атакой, а раскрытие ключа без использования специальных методов – компрометацией ключа. Успешная криптографическая атака называется взломом шифра. Надежность или криптостойкость симметричных и асимметричных алгоритмов зависит главным образом от ключей, а не от самих алгоритмов.

Голландский криптограф О. Керкгоффс первым сформулировал постулаты для оценки стойкости шифра перед его взломом. Согласно им, механизм преобразования считается известным злоумышленнику, а криптографическая стойкость алгоритмов преобразования определяется только неизвестным значением ключа.

При проведении атаки злоумышленник преследует две цели: найти открытый текст, зная шифротекст, но не имея ключа, либо найти сам закрытый ключ. Получив секретный ключ, злоумышленник полностью раскроет алгоритм шифрования.

В современном криптоанализе проводится различие между текстовой атакой без доступа к шифровальному устройству и текстовой атакой с доступом к шифровальному устройству. В зависимости от данных, которые может получить криптоаналитик, выделяют следующие типы атак:

1) *атаки на основе известного шифротекста* – у криптоаналитика есть шифротексты нескольких сообщений, зашифрованных одним алгоритмом, и его задача состоит в том, чтобы расшифровать сообщения или определить ключ;
2) *атаки на основе известного открытого текста* – у криптоаналитика есть зашифрованные тексты нескольких сообщений и открытый текст тех же сообщений, и его задача – идентифицировать ключ;
3) *атака на основе выбранного открытого текста* предполагает не только возможность криптоаналитика получить доступ к шифротекстам нескольких сообщений и открытым текстам тех же сообщений, но и способность выбирать открытый текст для шифрования. Задача состоит в том, чтобы определить ключ;
4) *атака, основанная на выбранном зашифрованном тексте*, позволяет криптоаналитику выбирать различные зашифрованные тексты для расшифровки. Он также имеет доступ к расшифрованным текстам. Атака в основном используется против асимметричных алгоритмов. Задача состоит в том, чтобы определить ключ;
5) *атаки на основе выбранного ключа* – криптоаналитик что-то знает о соединениях ключей;
6) *атака с адаптивным открытым текстом* – когда криптоаналитик может выбрать открытый текст несколько раз и получить соответствующий зашифрованный текст прямо во время атаки.

Сложность атаки можно оценить по следующим трем критериям:

- по сложности данных – оценивается количество данных, необходимых для проведения атаки;
- по сложности обработки – оценивается время, необходимое для реализации атаки;
- по требованиям к памяти ЭВМ – оценивается минимально необходимый объем памяти ЭВМ для выполнения всех расчетно-аналитических операций.

В дополнение к вышесказанному приведем одну из классификаций способов взлома с целью получения ключа (ключей):

- *Взлом с использованием грубой силы*;
- *Взлом с использованием анализа* (аналитическая атака). Изучение алгоритма проводится с целью выявления его слабых мест;

- *Взлом с использованием статистических данных* (статистическая атака). Архитектурные недостатки алгоритма устанавливаются путем определения статистических показателей шифротекста;
- *Атака на реализацию*. Он основан на надежности аппаратной и программной реализации алгоритма. В 1995 году было обнаружено, что при реализации алгоритма шифрования, используемого в браузере Netscape, начальным значением является системное время. С помощью системы с подобным генератором удалось сгенерировать необходимые ключи и взломать код;
- *Атака по времени*. На основе измерения времени, необходимого для расшифровки сообщения.

Ларс Кнудсен классифицировал сложность взлома алгоритмов по нескольким критериям: полная атака, глобальная дедукция, случайная (частичная) дедукция, информационная дедукция [73]. Кнудсен известен знаменитыми атаками на шифры SAFER и SQUARE, а также своими работами по криптоанализу.

Для взлома криптографического алгоритма при определенных условиях требуется определенное количество ресурсов: количество информации, необходимое для проведения атаки; время, необходимое для проведения атаки, и объем памяти, необходимый для хранения информации, используемой в атаке. Лучшей будет считаться атака, требующая минимального набора затрачиваемых ресурсов. Именно это характеризует криптостойкость криптографического алгоритма.

Таким образом, алгоритм считается криптографически стойким, если:

1) нет других методов взлома, кроме метода грубой силы;
2) размер ключа алгоритма достаточно велик, чтобы использование метода полного перебора стало невозможным на современном уровне развития вычислительной техники.

При сравнении двух и более алгоритмов определяется запас прочности. Для этого анализируется алгоритм с усеченным числом раундов (модифицированный). Запас криптографической стойкости можно определить, как отношение исходного количества раундов исследуемого алгоритма к максимальному количеству раундов его модификаций. Сравнительный анализ криптостойкости алгоритмов симметричного шифрования можно найти в работе В.О. Берникова [74]. В этой работе дана количественная оценка стойкости алгоритмов по следующим критериям: криптостойкость, запас прочности, скорость

расширения ключа, защита от атак по времени выполнения, реализация лавинного эффекта, возможность быстрого расширения ключа и возможность параллельных вычислений.

Криптографический алгоритм считается абсолютно устойчивым, если восстановление открытого текста невозможно для любого количества шифротекста. Такой алгоритм реализован в шифре Вернама и одноразовом блокноте, которые широко использовались разведчиками, военными и дипломатами разных стран. Классический одноразовый блокнот представляет собой неповторяющийся случайный набор ключей. Каждый ключ используется только один раз и только в одном сообщении. Второй важной особенностью является то, что длина ключа не должна быть меньше длины шифруемого сообщения. В настоящее время шифр Вернама применяется довольно редко, однако абсолютно стойкие шифры Вернама нашли практическое применение для защиты особо важных линий связи при относительно небольшом объеме информации.

Все остальные криптосистемы можно взломать, используя только шифротекст, перебирая все возможные ключи и проверяя осмысленность полученного открытого текста, то есть методом грубой силы. Время, необходимое для атаки методом полного перебора, зависит от двух параметров: количества проверяемых ключей и времени проверки одной комбинации ключей. Атаки грубой силы являются наиболее универсальными, но и самыми продолжительными. Оценка криптографической стойкости шифра основана на вычислительной сложности атаки грубой силы. Однако известны случаи успешного использования этой атаки. Например, в 2010 году на конференции DEFCON18 был представлен беспилотный летательный аппарат WASP, предназначенный для массового сбора статистики о домашних сетях Wi-Fi. Одной из его особенностей была возможность автоматического взлома беспроводных паролей методом грубой силы. Задача поиска секретного ключа шифрования методом полного перебора хорошо распараллеливается и может быть легко реализована для многопроцессорных вычислительных систем. Например, суперкомпьютеру Summit 2019 (OLCF-4) от IBM с производительностью 122,3 петафлопс требуется примерно в 7800 раз больше возраста Вселенной, чтобы перебрать все ключи (длина ключа составляет 128 бит, что означает 2128 возможных комбинаций). Самым мощным суперкомпьютером на июнь 2021 года стал Fugaku в Центре вычислительных наук Института физико-химических исследований (RIKEN) в Японии. Скорость выполняемых им вычислений составляет 442,01 петафлопс (1015 вычислительных операций с плавающей запятой в секунду). По этому показателю он в 3 раза быстрее и в 3 раза эффективнее предыдущего рекордсмена – Summit.

Рассмотрим некоторые из наиболее распространенных методов атак. Диффи и Хеллман в 1977 году предложили атаку типа «a meet-in-the-middle» (встреча посередине) – класс атак на криптографические алгоритмы, которые асимптотически сокращают время перебора с помощью «divide and conquer» и увеличивают требуемый объем памяти [75]. Примером может служить атака на семейство шифров KTANTAN в 2011 году [76], которая позволила уменьшить вычислительную сложность алгоритма вывода ключа с использованием атаки грубой силы с 280 до 272,9. Метод встречи посередине применим к алгоритмам шифрования, в которых используются два разных ключа, т. е. секретные подключи появляются через равные промежутки времени. Этот метод также применим для двойного шифрования данных, то есть сначала данные были зашифрованы с использованием секретного ключа, а затем полученный результат шифрования был снова зашифрован другим ключом. В качестве примера работы метода можно рассмотреть варианты анализа алгоритма двойного DES [77].

Атака с использованием связанного ключа – это тип криптографической атаки, при которой криптоаналитик выбирает связь между парой ключей, но сами ключи остаются ему неизвестными. Сопротивление таким атакам является важной целью разработки блочных шифров, и фактически это было одной из заявленных целей разработки алгоритма Rijndael. Одним из потенциальных применений таких атак является анализ хэш-функции. Особенно опасны атаки с использованием связанного ключа, если для построения функций хэширования используются алгоритмы симметричного шифрования.

Одной из важнейших качественных характеристик алгоритмов симметричного шифрования является безопасное время, определяющее устойчивость криптоалгоритма к использованию метода грубой силы для подбора ключа. Если длина ключа n бит, то для реализации этого метода необходимо провести около $2n$ операций криптоанализа. Однако существуют методы, позволяющие исключить из рассмотрения значительную часть вариантов значений ключей, что приводит к снижению вычислительной сложности криптоанализа по сравнению с перебором. Одним из таких методов является дифференциальный криптоанализ [78].

Дифференциальный криптоанализ был предложен в 1990 году Э. Бихамом и А. Шамиром для анализа алгоритма DES. Этот метод оказался первым методом взлома DES с усилием около 237, но с 247 вариантами выбранного открытого текста [79]. Дальнейшее развитие этого метода показало возможность его применения к целому классу различных типов шифров, позволило выявить слабые места многих используемых и разрабатываемых

алгоритмов шифрования. Сегодня этот метод, а также некоторые его производные, такие как линейный дифференциальный метод, метод невозможных дифференциалов (применяемый ко многим усеченным версиям шифров), метод бумеранга и его модификации, атака усиленного бумеранга и атака прямоугольником широко используются для оценки стойкости вновь созданных шифров.

На основе метода дифференциального криптоанализа разработаны последовательные алгоритмы поиска правильных пар текстов и секретного ключа для анализа n -раундового $(n \leq 16)$ алгоритма DES. Разработан параллельный алгоритм дифференциального анализа алгоритма Rijndael, лежащего в основе стандарта шифрования данных AES. Рассмотрена возможность применения метода дифференциального криптоанализа к анализу потоковых шифров и современных функций хэширования [80, 81]. Более подробную информацию о дифференциальном криптоанализе можно найти в [61].

Дифференциальный криптоанализ может взломать следующие алгоритмы: RC2 [82], RC5 [83], Thin-ICE [84], GDES [85].

Применение метода дифференциального криптоанализа на практике ограничено высокими требованиями ко времени и объему данных. Его можно использовать для взлома известного открытого текста, но в случае полного 16-раундового DES это делает его еще менее эффективным, чем атака грубой силой. Метод требует большого объема памяти для хранения возможных ключей. Эффективность метода также сильно зависит от структуры S-блоков взломанного алгоритма.

Линейный криптоанализ был изобретен М. Мацуи в 1993 году. Этот метод криптоанализа использует линейные приближения для описания работы шифра и, наряду с дифференциальным криптоанализом, является одним из наиболее распространенных методов взлома блочных шифров, но он также применим к поточным шифрам. Мацуи показал, что функцию шифрования можно представить в виде системы уравнений, которые имеют место с некоторой вероятностью. Существенным недостатком метода является необходимость иметь в наличии большой объем данных, которые шифруются с использованием одного и того же секретного ключа. В отличие от дифференциального криптоанализа, в котором высокое значение вероятности гарантирует успех атаки, в линейном криптоанализе успех анализа может быть обеспечен как уравнениями с очень высокой вероятностью, так и уравнениями с очень малой вероятностью. Кроме того, этот метод довольствуется известными открытыми текстами, что значительно расширяет область его применения. Помимо DES и FEAL линейному криптоанализу подлежат

следующие алгоритмы: RC5, если требуемый ключ шифрования из класса слабых ключей [86]; NUSH и NOEKEON [87, 88]. Знание механизмов работы метода линейного криптоанализа позволяет криптографам обеспечивать стойкость шифров еще на этапе проектирования криптоалгоритмов. Подробнее о линейном криптоанализе различных алгоритмов блочного шифрования можно прочитать в работе [89].

Суть методов алгебраического анализа заключается в получении уравнений, описывающих нелинейные преобразования замены *S*-блоков, с последующим решением найденных систем уравнений и получением ключа шифрования. В криптоанализе разработаны различные подходы к решению нелинейных систем булевых уравнений. Наиболее эффективными, как показывает практика криптоанализа, являются методы, использующие линеаризацию исходной системы. Например, **XL**-метод (e**X**tended **L**inearization – расширенная линеаризация) был предложен в работах Н. Куртуа, А. Климова, Ж. Патарина и А. Шамира [90]. Сложность анализа заключается в построении системы всевозможных линейных уравнений и последующем ее решении. Дополнительную информацию об алгебраическом криптоанализе можно найти в [91-94].

С появлением нового стандарта шифрования AES, в основе которого лежит алгоритм шифрования Rijndael, стал рассматриваться новый принцип построения блочных шифров. SQUARE – это симметричный блочный криптоалгоритм, разработанный в 1997 году В. Райменом, Д. Дейменом и Л. Кнудсеном. Структура алгоритма выбрана авторами из-за возможности получения эффективной реализации на широком диапазоне процессоров, а также из-за криптографической стойкости к дифференциальному и линейному криптоанализу. Первый криптоанализ SQUARE был проведен с использованием интегрального криптоанализа, который позже стал известен как атака SQUARE. В 2011 году был проведен криптоанализ полнораундного варианта SQUARE с использованием полного двудольного графа. Этот тип атаки позволил взломать шифр с помощью одного ключа, 248 открытых текстов и 2126 операций шифрования.

С увеличением скорости современных компьютеров высокоскоростные алгоритмы шифрования стали использовать все больше и больше раундов, признавая, что все существующие криптоаналитические технологии бесполезны. Если алгоритм шифрования имеет большое количество раундов, каждый дополнительный раунд требует экспоненциального увеличения усилий злоумышленника. Для криптоаналитика становится естественным поиск новых методов анализа, не зависящих от количества раундов в алгоритме шифрования. Этот новый метод был предложен в 1999 году А. Бирюковым и

Д. Вагнером и назван «Slide Attacks» [95]. Этот метод применим ко всем алгоритмам блочного шифрования. В то время как линейный и дифференциальный криптоанализ концентрируются в основном на общих свойствах методов шифрования, скользящая атака использует степень самоподобия, которая является фундаментальной разницей. Самоподобие означает использование одной и той же криптографической F-функции, которая зависит от одного и того же подключа в каждом раунде шифрования. Более изощренные варианты этой атаки имеют более сложный анализ и от них намного сложнее защититься [77].

В настоящее время в современной криптографии существуют следующие проблемы:

- Ограниченное количество рабочих схем. Современные криптографические алгоритмы основаны на определенной, пока неразрешимой проблеме, поэтому количество алгоритмов с открытым ключом невелико;
- Непрерывное «раздувание» размера блока данных и размера ключа из-за достижений в области математики и вычислительной техники. На момент создания криптосистемы RSA достаточен был размер чисел в 512 бит, сейчас же он достиг не менее 4 Кб, а в традиционной криптографии этот размер увеличился всего вдвое;
- Потенциально ненадежная основа. В рамках теории вычислительной сложности исследуется возможность решения сложных задач за полиномиальное время и доказывается связь большинства используемых вычислительно сложных задач с другими подобными задачами. Это означает, что взлом одной современной криптосистемы повлечет за собой взлом большинства остальных;
- Отсутствие дальней перспективы. Квантовые вычисления пока существуют только теоретически, но уже появляются практические достижения этой теории. Появление серьезных квантовых компьютеров существенно повлияет на современную криптографию.

Можно заключить, что для современной криптографии важным является повышение стойкости и уменьшение размера блоков данных путем модификации существующих криптосистем.

В современном мире криптография находит множество различных применений: в сотовой связи, цифровом телевидении, при подключении к Wi-Fi, на транспорте для защиты билетов от подделок, в банковских операциях, для защиты электронной почты от спама и во многих других областях [96].

1.5. Квазигруппы в криптологии

Современная алгебра, понимаемая как изучение операций над какими-либо математическими объектами, является одним из разделов математики, формирующим общие понятия и методы для всей математики. Отвлекаясь от природы объектов, но фиксируя некоторые свойства операций над ними, ученые пришли к понятию универсальной алгебры. В ходе развития математики и ее приложений первоначально выделялось сравнительно немного типов универсальных алгебр: группы, векторные пространства, ассоциативные кольца и алгебры, модули. Позднее предметом изучения стали другие классы: неассоциативные кольца и алгебры (алгебры Ли, Йордановы алгебры), решетки, полугруппы, квазигруппы, лупы и другие. Изучение установленных типов универсальных алгебр называется общей алгеброй.

В современной алгебре теорию квазигрупп можно рассматривать как одно из звеньев между классическими алгебраическими системами – группами и общими системами универсальной алгебры. Эта теория представляет собой самостоятельный раздел общей алгебры со своими задачами и проблемами. Квазигруппы весьма удобный объект для проверки гипотез и идей универсальной алгебры.

Теория квазигрупп в настоящее время имеет тесные связи с геометрией (теория проективных плоскостей), с комбинаторикой (теория латинских квадратов), с теорией алгебраических сетей. Эта теория возникла в 30-х годах XX века, когда появились работы Руф Муфанг, посвященные недезарговым проективным плоскостям. Квазигруппы имеют различные приложения в дифференциальной геометрии [97-99], теории автоматов [100], криптографии [7, 9, 10], физике [101] и т.д. Квазигруппы нашли свое применение в теории относительности при изучении пространственно-временных задач и появились такие понятия, как квазигруппа Пуанкаре и квазигруппа Лоренца [101]. С одной стороны, в недрах проективной геометрии возникли квазигруппы, а с другой, гораздо раньше – как комбинаторный объект – латинские квадраты в работах Леонарда Эйлера [7, 9, 10].

А.К. Сушкевич, один из основных основоположников теории обобщенных групп [102], исследовал бинарные квазигруппы с некоторыми дополнительными условиями (постулатами Сушкевича) и определил медиальные квазигруппы. К. Бурстин и В. Майер [103] изучали дистрибутивные квазигруппы. Автор этой работы принимал непосредственное участие в вычислении числа группоидов третьего порядка с точностью до изоморфизма с некоторыми тождествами типа Бола-Муфанг [104, 105].

Фундаментальные результаты в теории бинарных и n-арных квазигрупп, а также в теории сетей и теории функциональных уравнений принадлежат В.Д. Белоусову.

Важную роль в теории квазигрупп играет понятие изотопии, заимствованное А.А. Албертом из топологии, обобщающее понятие изоморфизма, которое используется в теории неассоциативных тел [106].

В классе квазигрупп, изотопных группам, интерес представляют так называемые линейные квазигруппы, которые были введены В.Д. Белоусовым в [107]. По аналогии с линейными квазигруппами были определены алинейные квазигруппы [108]. Позже, как обобщение линейных и алинейных квазигрупп, были введены классы квазигрупп линейных слева или справа, алинейных слева или справа и смешанного типа линейности.

Многие известные (классические) объекты относятся к классу обобщенных линейных квазигрупп [109]. Помимо результатов, полученных В.Д. Белоусовым [110], чешские алгебраисты – Т. Кепка, П. Немец, Я. Ежек и представители квазигрупповой школы В.Д. Белоусова – Г.Б. Белявская и ее ученики В.А. Щербаков, В.И. Избаш, К.К. Щукин, Ф.Н. Сохацкий, П.Н. Сырбу, А.Х. Табаров, В. А. Дудек провели исследования по линейным квазигруппам и их обобщениям. Исследованы алгебраические и комбинаторные аспекты обобщенных линейных квазигрупп. В последнее время появился интерес к изучению обобщенных квазигрупповых производных [111-113].

Почти все известные конструкции кодов обнаружения и исправления ошибок, криптографические алгоритмы и системы шифрования использовали ассоциативные алгебраические структуры, такие как группы и поля [5, 6]. Было показано, что такие неассоциативные структуры, как квазигруппы и неополя, можно использовать во многих разделах теории кодирования и особенно в криптологии. Коды и шифры на основе неассоциативных систем демонстрируют лучшие возможности, чем известные коды и шифры на основе ассоциативных систем [8].

В последние годы интенсивно развиваются квантовая теория кода и квантовая криптология [114-116]. Важно отметить, что квантовая криптология использует теоретические достижения классической криптологии [117].

Эффективность применения квазигрупп в криптологии основана на том, что квазигруппы представляют собой «обобщенные перестановки» того или иного вида и число квазигрупп порядка n больше, чем$(n! \cdot (n-1)! \cdot \ldots \cdot 2! \cdot 1!)$ [9]. Использование квазигрупп в криптологии дает такие же перестановки и замены, но они легче генерируются и не

требуют большого объема памяти устройства, воздействуя «локально» только на один блок открытого текста.

Поскольку латинский квадрат имеет комбинаторную природу и представляет собой таблицу умножения квазигруппы, его можно рассматривать как самое раннее использование неассоциативной алгебраической структуры в криптологии. Впервые в криптографии латинский квадрат был использован в полиалфавитном шифре Тритемия. Впоследствии этот шифр был усовершенствован Дж.Б. Белласо. Вместе с идеей Л.Б. Альберти использовать произвольный алфавит, появился новый шифр на основе квазигруппы, что стало важной вехой в развитии криптографии [13].

Имеется возможность развития этого направления с использованием ортогональных систем бинарных или n-арных квазигрупп. Построение линейных двоичных кодов с использованием ортогональных систем латинских квадратов можно найти в [118].

Рудольф Шауфлер в своей диссертации обсудил минимальное количество открытого текста и соответствующего ему зашифрованного текста, которые потребуются для взлома шифра Виженера [119]. Он рассмотрел минимальное количество заполнений частичного латинского квадрата, которое полностью определяет квадрат. В последнее время эта проблема вновь возникла как проблема определения так называемых критических множеств в латинских квадратах [120-123]. Во время Второй мировой войны Р. Шауфлер, работая в немецкой криптографической службе, разработал метод обнаружения ошибок, основанный на использовании обобщенных тождеств (название было дано позднее В. Д. Белоусовым), в котором контрольные цифры вычисляются с помощью ассоциативной системы квазигрупп. Сведения о системах квазигрупп с обобщенными тождествами, применяемых в современной криптографии, можно найти в работах Ю. М. Мовсисяна [14].

Поточные шифры более подходят, а в некоторых случаях даже обязательны, когда буферизация ограничена или, когда символы должны обрабатываться индивидуально по мере их получения [124]. Часто предыдущий зашифрованный блок используется для шифрования блока открытого текста, и мы уже говорили, что Фейстель был одним из первых, кто предложил этот метод шифрования (сеть Фейстеля) [125].

Среди примеров использования латинских квадратов для построения потоковых шифров необходимо выделить предложенный в 2005 году шифр Edon-80. Разработчики шифра из 576 существующих латинских квадратов 4-го порядка тщательно отобрали 4, на основе которых в криптосхему встроен конвейер из 80 латинских квадратов, он используется для генерации гаммы [13].

Квазигрупповые понятия, такие как изотопия, квазигрупповая операция, используются при построении стандарта блочного шифрования в [126]. Ч. Кошельны в своих работах показал, как можно создавать поточные шифры на основе квазигрупп (неополей), которые более эффективны и надежны, чем на основе групп (полей) [127, 128].

В [15] авторы строят криптосистему, используя такую неассоциативную структуру, как кольцо квазигрупп. Там же описываются модификации, повышающие защищенность этой схемы от возможных атак, анализируются некоторые неассоциативные структуры, пригодные для построения криптосистем.

Криптография с открытым ключом, основанная на полугрупповых действиях, изучается в [16]. Двустороннее групповое действие используется для криптологической необходимости в [17]. Криптографически пригодные квазигруппы, основанные на функциональных уравнениях, изучаются в [19]. Криптосистема с открытым ключом, использующая обобщенные поточные шифры на основе квазигрупп, представлена в [24].

Под идентификацией сообщения мы подразумеваем, что получатель сообщения может установить его происхождение, а злоумышленник не может маскироваться под кого-то другого. Некоторые квазигрупповые подходы к проблемам идентификации сообщений, получения динамического пароля, цифровых отпечатков пальцев обсуждаются в [129].

Под аутентификацией сообщения мы подразумеваем, что получатель сообщения может убедиться, что сообщение не было изменено при передаче. Авторы предложили новую схему аутентификации на основе квазигрупп в [130]. Дополнительную информацию по этому вопросу можно найти в [10, 131]. В [21] создан новый код аутентификации сообщений QMAC, безопасность которого основана на неассоциативности квазигрупп.

В [132] рассмотрено несколько криптосистем, основанных на квазигруппах и различных комбинаторных объектах. Изотопию квазигрупп в протоколе с нулевым разглашением предлагается использовать в [133].

При построении криптосистем на основе квазигрупп возникает вопрос: насколько велико расстояние между различными бинарными или n -арными квазигруппами? Информация о расстоянии Хэмминга между операциями квазигруппы дана в работах [134-136].

Важной криптографической задачей является генерация «больших» квазигрупп, которые можно легко хранить в компактном виде в памяти компьютера. Понятно, что для этих целей наиболее подходящим способом является сохранение небольшой базы. Поэтому возникла необходимость легко генерировать объекты (циклические группы, абелевы

группы) с помощью быстрых методов их преобразования (парастрофы, изотопии, изострофы [137], гомотопии, обобщенные изотопии [138], скрещенное произведение, обобщенное скрещенное произведение). Для этих целей вполне подходят различные линейные квазигруппы (особенно n-арные квазигруппы) [139, 140, 109].

В [141] предлагается использовать булевы векторы и простые поля для построения n -арных и бинарных квазигрупп. Метод генерации почти неограниченного числа квазигрупп произвольного порядка с помощью системы компьютерной алгебры Maple7 представлен в [128]. Авторы дают эффективный алгоритм построения огромных квазигрупп с использованием расширенных сетей Фейстеля в [142].

Квазигруппы нашли свое применение в схемах секретного доступа. Существуют схемы разделения секретов, использующие китайскую теорему об остатках (схемы Миньо и Асмута-Блума) и ортогональные массивы. Схемы разделения секретов, основанные на критических множествах в латинских квадратах, изучены в [143]. Исследование недостатков некоторых теоретических криптографических схем, основанных на критических множествах в латинских квадратах, продолжается в [144]. Открытые вопросы, касающиеся схем разделения секретов с использованием латинских квадратов, можно найти в [145-147]. Г.Б. Белявская предлагает общую схему разделения секретов, основанную на частично ортогональных системах k-арных операций, которая обобщает некоторые известные схемы [148].

Некоторые применения CI -квазигрупп (скрещенных инверсных квазигрупп) в криптологии с асимметричными ключами описаны в [149]. CI-квазигруппа может быть использована для предоставления одноразового обмена ключами (без вмешательства центра распределения ключей) [150]. В этой схеме В.А. Щербаков предлагает применять (r, s, t)-инверсные квазигруппы [29]. Схему можно обобщить, используя некоторые m-инверсные квазигруппы [151], (r, s, t)-инверсные квазигруппы [152, 153] или (α, β, γ)-инверсные квазигруппы [154].

В [25, 26] предлагается использовать квазигруппы для безопасного кодирования. С. Марковски и его соавторы представили в [26] поточный шифр с почти открытым ключом, основанный на квазигруппах. Показано, что ключ может быть открытым и при этом иметь достаточную защищенность.

1.6. Алгоритм Марковского и его обобщения

Алгоритм Марковского и его обобщения в настоящее время широко известны и часто используются для построения поточных шифров на основе квазигрупп. В этом вопросе рассматриваются основные обобщения, которые были получены для алгоритма Марковского. Сам алгоритм Марковского и подробное описание его работы можно найти во второй главе данной работы (Алгоритм 2.2.1), а также в [29].

В классическом алгоритме Марковского квазигруппа $(Q, \cdot)$ и её (23)-парастроф $(Q, \backslash)$ удовлетворяют двум тождествам: $x \backslash (x \cdot y) = y,\ x \cdot (x \backslash y) = y,$ которые используются для построения потокового шифра.

Некоторые важные результаты были получены в [30]. Авторы находят распределение k-кортежей букв после n применений квазигруппового преобразования $(k > n)$, то есть алгоритма Марковского, а также приводят алгоритм статистической атаки для обнаружения исходного сообщения и дают рекомендации по защите оригинальных сообщений.

А. Крапеж и Д. Живкович определяют парастрофические преобразования квазигрупп в [33]. Эти преобразования весьма перспективны для дальнейшего исследования и применения [31].

Согласно классическому эквациональному определению бинарной квазигруппы (Определение 2.1.3) должны выполняться тождества:

$$A({}^{(13)}A(x, y), y) = x \quad \textbf{(1.1)}$$

$${}^{(13)}A(A(x, y), y) = x \quad \textbf{(1.2)}$$

$$A(x, {}^{(23)}A(x, y)) = y \quad \textbf{(1.3)}$$

$${}^{(23)}A(x, A(x, y)) = y\,. \quad \textbf{(1.4)}$$

Операцию A часто обозначают " $\cdot$ ", операцию ${}^{(23)}$A как "\" и операцию ${}^{(13)}$A как "/".

В алгоритме Марковского также можно использовать квазигруппу (Q, A) и её (13), (123) и (132) парастрофы, и для них будут выполняться такие тождества, как (1.2), (1.5) и (1.6) соответственно [155, 31, 32].

$${}^{(123)}A(A(x, y), x) = y, \quad \textbf{(1.5)}$$

$${}^{(132)}A(y, A(x, y)) = x. \quad \textbf{(1.6)}$$

Результаты, полученные в [27] подтверждают возможность использования квазигрупп в криптографии. Авторы утверждали, что построенный шифр устойчив к атакам грубой силы и статистическим атакам. Позже аналогичные результаты были представлены

в [25]. Результат реализации построенной схемы основан на использовании квазигруппы порядка 256, а фрагмент кода процедур шифрования и дешифрования также представлен в [25]. В диссертационной работе М. Войводы [34] доказано, что этот шифр не устойчив к атаке с помощью выбранного шифротекста и выбранного открытого текста. Утверждается, что этот шифр не устойчив к специальной статистической атаке.

Есть несколько способов обобщить алгоритм Марковского. Наиболее очевидный способ – увеличить арность квазигруппы, т. е. вместо бинарных используются n-арные ($n \geq 3$) квазигруппы. Этот метод был предложен В.А. Щербаковым в [35, 36] и реализован А. Петреску используя язык Ассемблер [37, 38]. Шифрование и дешифрование выполняется быстро. А еще шифр характеризуется хорошей устойчивостью к статистическим атакам [37]. Автор предлагает два n-квазигрупповых симметричных поточных шифра в [38].

Некоторые модификации, которые делают алгоритм Марковского более устойчивым к известным атакам, можно найти в [32]. Одна из таких попыток с учетом результатов Войводы [34] была предложена В.А. Щербаковым в [39]. Алгоритм строится с использованием тернарной квазигруппы и парастрофических преобразований. Дополнительно автор предлагает использовать систему n-арных ортогональных операций.

Дальнейшее развитие алгоритма Марковского представлено в [40]. Авторы предлагают Edon-R, который представляет собой класс хэш-функций с переменной выходной длиной, которая определяется с помощью квазигрупп и квазигрупповых преобразований строк.

Блочный шифр, основанный на алгоритме Марковского, предложен в [41]. Авторы построили криптосистему с открытым ключом MQQ, используя квазигруппы.

В [25] предложено использовать алгоритм Марковского для безопасного шифрования файловой системы. Основная идея состоит в том, чтобы представить квазигруппы в виде векторозначных булевых функций, чтобы найти класс квазигрупп со степенью не выше 2, а затем использовать квазигрупповые преобразования строк для построения биективных многомерных квадратичных полиномов.

Большинство мобильных операторов шифруют все мобильные данные, включая SMS-сообщения. Эти требования требуют разработки дополнительного шифрования для SMS-сообщений, чтобы общаться могли только аккредитованные стороны. Подход к этой проблеме с использованием алгоритма Марковского описан в [42].

В работе В.А. Щербакова [29] алгоритм Марковского был переписан с использованием левых трансляций. Далее автор предлагает следующее обобщение: вместо

трансляций можно использовать степени этих трансляций в шифровальной части этого алгоритма. Степени левых трансляций должны меняться от шага к шагу, чтобы защитить этот алгоритм от атак выбранным открытым текстом и зашифрованным текстом. Правило получения степеней должно быть известно получателю. Предлагается использовать ряд правил выбора степеней трансляций [29] и правые и средние трансляции [35] вместо левых трансляций.

N-арный аналог алгоритма Марковского был построен в [38, 29]. В процедуре шифрования и в процедуре дешифрования может использоваться более одной n-квазигрупповой операции.

В статье [43] предлагаются следующие обобщения алгоритма Марковского:

1) Предлагается использовать композицию бинарных квазигрупп для построения некоторых n-арных $(n \geq 2)$ квазигрупп;
2) Предлагается использовать изотопию n-арных квазигрупп для порождения ключевой квазигруппы.

Модифицированный поточный шифр на основе тернарных квазигрупп предложен в [43]. Схема предназначена для экспоненциального увеличения сложности ключа.

Важные сведения о криптоанализе некоторых поточных шифров можно найти в статье В.А. Щербакова и П. Ксорго [44]. Бинарный аналог этой атаки описан М. Войводой в [34].

Возможно использование системы ортогональных n-арных группоидов в качестве дополнительной процедуры построения почти поточного шифра [39]. Такие системы имеют более равномерное распределение элементов базового набора и поэтому могут быть более предпочтительными в плане защиты от статистических криптоаналитических атак. Процедуру шифрования можно найти в [29]. Она основана на использовании схемы Фейстеля [125]. Дешифровка алгоритма основана на том факте, что ортогональная система из n n-арных операций имеет единственное решение для любого набора элементов $a_1, a_2, \dots, a_n$. Однако применение только одного шага алгоритма не очень безопасно, так как эта процедура не устойчива к атакам с использованием выбранного зашифрованного текста и выбранного открытого текста. Можно, следуя «векторным идеям» [156], в качестве первого шага записать любую букву открытого текста u_i в виде n- кортежа, а затем применить алгоритм. Можно использовать бинарное представление символов алфавита A. Если в системе ортогональных n-арных операций имеется хотя бы одна n-арная квазигруппа, то можно применить алгоритм Марковского в сочетании с последним

алгоритмом с некоторой непериодической частотой. Предлагаемые модификации усложняют реализацию атак выбранным открытым текстом и выбранным зашифрованным текстом [29].

В.А. Щербаков предлагает использовать одновременно алгоритм Марковского, построенный для n-арного случая с обратимостью, на последнем месте в сочетании с алгоритмом шифрования, основанным на системах ортогональных n-арных операций. Одно из возможных обобщений реализовано в [29].

Для построения обобщенного алгоритма Марковского для n-арного случая с обратимостью на последнем месте (в том числе для алгоритма, использующего трансляции различной степени) предлагается использовать ортогональные системы бинарных парастрофических квазигрупп, в частности, ортогональные системы бинарных парастрофических T-квазигрупп. Для их построения используются GAP и Prover. Сравнение мощности алгоритма Марковского и алгоритмов, предложенных автором, приведено в [29].

Рональд Ривест ввел режим шифрования «все или ничего» (AON), чтобы усложнить поиск методом грубой силы путем соответствующей предварительной обработки сообщения перед его шифрованием [157]. Основная идея состоит в том, чтобы хранить ключ небольшой длины (например, 64-битный) для базового шифрования, которое может обрабатываться специальным оборудованием без достаточной вычислительной мощности или памяти. В статье [158] предлагается специальное преобразование, основанное на использовании квазигруппы, в том числе на использовании алгоритма Марковского.

Алгоритм Марковского можно использовать для построения аналогов схемы Эль-Гамаля. Классическая система шифрования Эль-Гамаля формулируется на языке теории чисел с использованием умножения по модулю простого числа, а сложность этой системы основана на сложности задачи дискретного логарифмирования [159]. Система шифрования Эль-Гамаля не защищена от атаки выбранным зашифрованным текстом [160]. Обычно она используется в гибридных криптосистемах, где само сообщение и ключ шифруются с использованием симметричной криптосистемы.

В [45] представлен аналог системы шифрования Эль-Гамаля на основе алгоритма Марковского. В этом алгоритме вместо изотопии также можно использовать изострофию [161] и вместо бинарных квазигрупп n-арные $(n > 2)$ квазигруппы [139, 39].

Протокол генерации общего секретного ключа на основе луп Муфанг, обобщение схемы Эль-Гамаля на лупы Муфанг и обобщение схемы Эль-Гамаля на основе

квазиавтоморфизмов квазигрупп приведены в [46]. Дискретно-логарифмическая задача о лупах Муфанг сводится к такой же задаче над конечными простыми полями в [162].

На квантовом компьютере существуют полиномиальные алгоритмы простой факторизации и дискретного логарифмирования [163]. Поэтому схемы RSA и Эль-Гамаля, их аналоги и многие обобщения не могут быть безопасными, если квантовым компьютером может воспользоваться нелегальный пользователь.

Гомоморфное шифрование – это свойство схемы шифрования, позволяющее выполнять вычисления с зашифрованным текстом без его расшифровки [164, 165]. Гомоморфное шифрование открывает новые возможности по сохранению целостности, доступности и конфиденциальности данных при их обработке.

Таким образом, существует возможность использования квазигрупп и неополей почти во всех разделах теории кодирования, и особенно в криптологии. Теория приложений квазигрупп в криптологии в настоящее время переживает период интенсивного развития и представляет собой весьма перспективную область.

1.7. Выводы по Главе 1

В данной главе представлен обзор особенностей современных криптографических систем, а также рассмотрены основные проблемы, связанные с определением криптографической стойкости современных систем защиты информации и подходы к их решению. Были выделены основные перспективы применения теории квазигрупп в криптологии, и, в частности, проведен обзор различных способов обобщения алгоритма Марковского и схемы Эль-Гамаля. Кроме того, осуществлен синтез представленных в литературе научных исследований по эволюции развития методов анализа для изучения теории квазигрупп, теории кодирования и изучения основных приложений квазигрупп в теории кодирования.

В первой главе освещается современная ситуация в области использования информационных технологий при разработке криптографических и алгебраических алгоритмов.

На основании анализа текущей ситуации можно сделать следующие выводы:

1) Важным аспектом безопасности информационных систем является оценка надежности используемых криптографических алгоритмов;
2) Ключевой задачей защиты информации является создание надежных алгоритмов шифрования;

3) Любой вновь построенный алгоритм необходимо подвергать тщательному анализу с целью выявления его слабых мест и возможности взлома;
4) Использование квазигрупп в криптологии показывает лучшие возможности и результаты, чем использование ассоциативных систем.

На основании вышеизложенного была сформулирована цель исследования, заключающаяся в изучении и построении новых модификаций и усовершенствовании уже построенных криптографических алгоритмов на основе алгоритма Марковского и проведении их криптоанализа.

Некоторые результаты автора, относящиеся к этой главе, были опубликованы в [48, 104, 105, 111-113, 118].

Для достижения поставленной цели были обозначены следующие задачи:

- разработка эффективного криптографического алгоритма на основе алгоритма Марковского с использованием квазигрупп;
- разработка программ, реализующих работу построенных алгоритмов;
- проведение атак на все изученные и построенные шифры;
- осуществление сравнительного анализа проведенных атак;
- подбор оптимальных вариантов текстов для всех изученных типов атак.

2. АЛГОРИТМ МАРКОВСКОГО И ЕГО НОВЫЕ ОБОБЩЕНИЯ

Сегодня разные точки зрения на одну и ту же математическую идею приводят к разным обобщениям.

Вторая глава посвящена разработке криптографических и алгебраических алгоритмов на основе алгоритма Марковского. Для построения обобщенных алгоритмов можно использовать такие алгебраические структуры, как квазигруппы (включая левую и правую квазигруппы) и группоиды, обратимые на одном фиксированном месте. Однако основная задача состоит не только в построении новых и модификации существующих алгоритмов, но прежде всего в выявлении особенностей их функционирования и сравнении степени стойкости обобщенных алгоритмов к различным типам атак, а также их программной реализации.

2.1. Основные понятия и определения

Начнем с основных понятий и определений, которые будут использоваться при построении алгоритма Марковского и его обобщений.

Определение 2.1.1. Непустое множество Q с определенной на нем бинарной операцией A называется *бинарным группоидом* и обозначается (Q, A).

На множестве Q существует столько бинарных группоидов, сколько на нем можно определить различных бинарных операций [139].

Определение 2.1.2. Группоид (Q, A) называется *квазигруппой*, если уравнения: $A(a, x) = b$ и $A(y, a) = b$ однозначно разрешимы $\forall a, b \in Q$.

Это определение квазигруппы называется *экзистенциальным* и эквивалентно следующему: группоид (Q, A) называется квазигруппой, если каждый из двух элементов из равенства $A(a, b) = c$ однозначно определяет третий элемент.

Существуют и другие определения квазигрупп [166].

Любой конечный группоид можно задать с помощью таблицы Кэли. Например:

Таблица 2.1. Таблица Кэли группоида $(G, *)$

*	**1**	**2**	**3**	**4**
1	3	4	2	4
2	4	2	1	1
3	1	2	3	3
4	2	3	4	1

Для квазигрупп таблица Кэли имеет особенность: все элементы в каждом столбце и в каждой строке различны. Например:

Таблица 2.2. Таблица Кэли квазигруппы$(G, \cdot)$

$\cdot$	**1**	**2**	**3**	**4**
1	4	3	1	2
2	3	2	4	1
3	2	1	3	4
4	1	4	2	3

В дополнение к экзистенциальному существует эквивалентное эквациональное определение квазигруппы [139, с.6-7]:

Определение 2.1.3. Группоид (Q, A) называется *квазигруппой*, если существуют две другие операции: B: $B(a, b) = x$ и C: $C(b, a) = y$, заданные на одном множестве Q, такие, что в алгебре $Q(A, B, C)$ выполняются тождества:

$$A[x, B(x, y)] = y,\ B[x, A(x, y)] = y,\ A[C(y, x), x] = y,\ C[A(y, x), x] = y.$$

Операция B называется *правой обратной* для A и обозначается: $B = A^{-1}$.

Операция C называется *левой обратной* для A и обозначается: $C = {}^{-1}A$.

Определение 2.1.4. Квазигруппа $(Q, \cdot)$ называется *лупой*, если существует элемент $e \in Q|$ $a \cdot e = e \cdot a = a, \forall a \in Q$, где $e -$ называется *единицей квазигруппы*.

Определение 2.1.5. Квазигруппа называется *группой*, если в $(Q, \cdot)$ выполняется ассоциативный закон: $xy \cdot z = x \cdot yz$.

Определение 2.1.6. *Операция B изотопна операции A*, если $\exists\, \alpha, \beta, \gamma -$ подстановки множества Q, такие, что: $B(x, y) = \gamma^{-1}A(\alpha x, \beta y) \quad \forall x, y \in Q, T = (\alpha, \beta, \gamma)$ –это *изотопия*, а $\alpha, \beta, \gamma -$ это *левая, правая и главная компоненты изотопии* соответственно.

Если в последнем определении $\alpha = \beta = \gamma,$ то изотопия становится *изоморфизмом*.

Определение 2.1.7. Упорядоченная тройка подстановок $T = (\alpha, \beta, \gamma)$ множества Q называется *автотопией* квазигруппы $(Q, \cdot)$, если: $\gamma^{-1}(\alpha x \cdot \beta y) = x \cdot y,\ \forall x, y \in Q$.

Автотопия – это частный случай изотопии, когда $B = A$.

Автоморфизм – это автотопия вида: $(\alpha, \alpha, \alpha) = \alpha$.

Определение 2.1.8. Любой выбранный фиксированный элемент l из некоторого непустого алфавита Q называется лидером (или элементом-лидером).

Определение 2.1.9. С заданной бинарной квазигруппой (Q, A) можно связать пять других, так называемых *парастрофов квазигруппы* (Q, A):

$A^{(12)}(x_2, x_1) = x_3$ (операция $*$),

$A^{(13)}(x_3, x_2) = x_1$ (операция $/$),

$A^{(23)}(x_1, x_3) = x_2$ (операция $\backslash$),

$A^{(123)}(x_2, x_3) = x_1$ (операция $//$),

$A^{(132)}(x_3, x_1) = x_2$ (операция $\backslash\backslash$).

Квазигруппа $(Q, \cdot)$ и ее (23)-парастроф $(Q, \backslash)$ удовлетворяют следующим двум тождествам: $x\backslash(x \cdot y) = y,\ x \cdot (x\backslash y) = y$. Предлагается использовать это свойство квазигрупп для построения поточных шифров, а именно, известного традиционного алгоритма Марковского.

2.2. Алгоритм Марковского

Алгоритм 2.2.1. Пусть $A -$ непустой алфавит, $k -$ натуральное число, $u_i,\ v_i \in A$ и $i \in \{1, \dots k\}$.

Определим квазигруппу $(A, \cdot)$. Ясно, что после этого квазигруппа $(A, \backslash)$ определится однозначным образом.

Зафиксируем элемент $l \in A$, который назовем лидером.

Пусть $u_1, u_2, \dots, u_k\ -k$- кортеж букв алфавита A.

С. Марковски предлагает воспользоваться для шифрования следующей процедурой:

$v_1 = l \cdot u_1$,

$v_2 = v_1 \cdot u_2$, …,

$v_i = v_{i-1} \cdot u_i$, где $i = 2,3, \dots k$.

В результате будет получен следующий зашифрованный текст: $v_1, v_2, \dots, v_k$.

В этой процедуре шифрования закрытым ключом как для отправителя, так и для получателя является квазигруппа $(A, \cdot)$ и лидер l.

Процедура дешифрования будет иметь следующий вид:

$u_1 = l\backslash v_1$,

$u_2 = v_1\backslash v_2$, …,

$u_i = v_{i-1}\backslash v_i$, где $i = 2,3 \dots k$.

В процедуре дешифрования закрытым ключом для получателя является квазигруппа $(A, \backslash)$ и тот же лидер l. Следовательно, это криптосистема с почти симметричными ключами.

Разработчик алгоритма утверждает, что построенный шифр устойчив к атаке методом полного перебора и к статистической атаке [29].

Пример 2.2.2. Пусть алфавит A имеет вид: $A = \{0,1,2,3\}$. Построим следующую квазигруппу $(A,\ \cdot)$:

Таблица 2.3. Таблица Кэли квазигруппы $(A,\ \cdot)$

$\cdot$	**0**	**1**	**2**	**3**
0	1	0	3	2
1	0	2	1	3
2	3	1	2	0
3	2	3	0	1

Квазигруппа $(A,\ \backslash)$ или $A^{(23)}(x_1, x_3) = x_2$ будет иметь точно такую же таблицу Кэли (Таблица 2.3), которая обладает симметрией относительно главной и побочной диагоналей.

Пусть $l = 3$ и открытый текст имеет вид: $u =\ 031323110$. Имеем:

$v_1 = l \cdot u_1 = 3 \cdot 0 = 2,$

$v_2 = v_1 \cdot u_2 = 2 \cdot 3 = 0,$

$v_3 = v_2 \cdot u_3 = 0 \cdot 1 = 0,$

$v_4 = v_3 \cdot u_4 = 0 \cdot 3 = 2,$

$v_5 = v_4 \cdot u_5 = 2 \cdot 2 = 2,$

$v_6 = v_5 \cdot u_6 = 2 \cdot 3 = 0,$

$v_7 = v_6 \cdot u_7 = 0 \cdot 1 = 0,$

$v_8 = v_7 \cdot u_8 = 0 \cdot 1 = 0,$

$v_9 = v_8 \cdot u_9 = 0 \cdot 0 = 1.$

Получаем следующий зашифрованный текст: $v = 200220001$.

Применяя функцию декодирования к v получим:

$u_1 = l \backslash v_1 = 3 \backslash 2 = 0,$

$u_2 = v_1 \backslash v_2 = 2 \backslash 0 = 3,$

$u_3 = v_2 \backslash v_3 = 0 \backslash 0 = 1,$

$u_4 = v_3 \backslash v_4 = 0 \backslash 2 = 3,$

$u_5 = v_4 \backslash v_5 = 2 \backslash 2 = 2,$

$u_6 = v_5 \backslash v_6 = 2 \backslash 0 = 3,$

$u_7 = v_6 \backslash v_7 = 0 \backslash 0 = 1,$

$u_8 = v_7 \backslash v_8 = 0 \backslash 0 = 1,$

$u_9 = v_8 \backslash v_9 = 0 \backslash 1 = 0.$

Результат совпадает с исходным открытым текстом $u = 031323110$.

Реализацию алгоритма Марковского для этого примера можно найти в Приложении 2: Программа А2.1 для шифрования и Программа А2.2 для дешифрования. Более того, полученные программы работают для шифрования и дешифрования текстов заданной длины (длину текста можно легко изменить). Программы работают для любого выбранного значения лидера.

Алгоритм Марковского будет работать для левой и правой квазигрупп. При этом для левых квазигрупп алгоритм не будет отличаться от описанного выше классического алгоритма.

Пример 2.2.3. Выберем алфавит вида: $A = \{0,1,2,3\}$. Построим левую квазигруппу, в которой однозначно разрешимо только одно уравнение: $A(a, x) = b$ и, поэтому в столбцах элементы могут повторяться, а в строках они не повторяются. Квазигруппы $(A, \cdot)$ и $(A, \backslash)$ имеют следующие таблицы Кэли:

Таблица 2.4. Таблица Кэли левой квазигруппы $(A, \cdot)$

$\cdot$	**0**	**1**	**2**	**3**
0	1	2	0	3
1	0	3	2	1
2	1	2	3	0
3	2	1	0	3

Таблица 2.5. Таблица Кэли левой квазигруппы$(A, \backslash)$

$\backslash$	**0**	**1**	**2**	**3**
0	2	0	1	3
1	0	3	2	1
2	3	0	1	2
3	2	1	0	3

Пусть $l = 3$ и открытый текст $u = 031323110$, тогда имеем:

$v_1 = l \cdot u_1 = 3 \cdot 0 = 2,$

$v_2 = v_1 \cdot u_2 = 2 \cdot 3 = 0,$

$v_3 = v_2 \cdot u_3 = 0 \cdot 1 = 2,$

$v_4 = v_3 \cdot u_4 = 2 \cdot 3 = 0,$

$v_5 = v_4 \cdot u_5 = 0 \cdot 2 = 0,$

$v_6 = v_5 \cdot u_6 = 0 \cdot 3 = 3,$

$v_7 = v_6 \cdot u_7 = 3 \cdot 1 = 1,$

$v_8 = v_7 \cdot u_8 = 1 \cdot 1 = 3,$

$v_9 = v_8 \cdot u_9 = 3 \cdot 0 = 2.$

Получили зашифрованный текст вида: $v = 202003132.$

Проверим результат с помощью процедуры дешифрования:

$u_1 = l \backslash v_1 = 3 \backslash 2 = 0,$

$u_2 = v_1 \backslash v_2 = 2 \backslash 0 = 3,$

$u_3 = v_2 \backslash v_3 = 0 \backslash 2 = 1,$

$u_4 = v_3 \backslash v_4 = 2 \backslash 0 = 3,$

$u_5 = v_4 \backslash v_5 = 0 \backslash 0 = 2,$

$u_6 = v_5 \backslash v_6 = 0 \backslash 3 = 3,$

$u_7 = v_6 \backslash v_7 = 3 \backslash 1 = 1,$

$u_8 = v_7 \backslash v_8 = 1 \backslash 3 = 1,$

$u_9 = v_8 \backslash v_9 = 3 \backslash 2 = 0.$

Реализацию алгоритма Марковского для этого примера можно найти в Приложении 2: Программа А2.3 для шифрования и Программа А2.4 для дешифрования. Эта программная реализация будет работать для любой левой бинарной квазигруппы. В результате была подтверждена корректная работа алгоритма Марковского для левых квазигрупп.

В следующем параграфе описывается работа алгоритма Марковского для правых квазигрупп и выявляются его отличия от классического алгоритма.

2.3. Алгоритм Марковского для правой квазигруппы

Пусть теперь $(Q, \cdot)$ – правая квазигруппа, для которой только одно уравнение однозначно разрешимо: $A(a, x) = b$, и для нее определен (13)-парастроф $(Q, /)$. Для этой квазигруппы и ее(13)-парастрофа выполняются следующие тождества:

$(y/x) \cdot x = y$ и $(y \cdot x)/x = y$.

Алгоритм Марковского для правой квазигруппы в случае *шифрования* будет иметь вид:

$v_1 = u_1 \cdot l,$

$v_2 = u_2 \cdot v_1, \ldots,$

$v_i = u_i \cdot v_{i-1}$, где $i = 2,3 \ldots k$ и $k -$ это длина шифруемого текста.

Алгоритм дешифрования строится следующим образом:

$u_1 = v_1/l,$

$u_2 = v_2/v_1, \ldots,$

$u_i = v_i/v_{i-1}$, где $i = 2,3 \ldots k.$

Пример 2.3.1. Пусть выбран алфавит: $A = \{0,1,2,3,4\}$. Построим следующую правую квазигруппу (элементы могут повторяться в строках, но не в столбцах) и ее (13)-парастроф:

Таблица 2.6. Таблица Кэли правой квазигруппы $(A, \cdot)$

$\cdot$	**0**	**1**	**2**	**3**	**4**
0	0	3	2	1	0
1	2	0	1	0	4
2	3	2	0	2	1
3	1	1	4	3	2
4	4	4	3	4	3

Тогда квазигруппа $(A, /)$ или $A^{(13)}(x_3, x_2) = x_1$ имеет следующий вид:

Таблица 2.7. Таблица Кэли правой квазигруппы $(A, /)$

$/$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	1	0
1	3	3	1	0	2
2	1	2	0	2	3
3	2	0	4	3	4
4	4	4	3	4	1

Пусть $l = 2$ и открытый текст $u = 031323110$. Тогда шифрование будет иметь вид:

$v_1 = u_1 \cdot l = 0 \cdot 2 = 2,$

$v_2 = u_2.v_1 = 3 \cdot 2 = 4,$

$v_3 = u_3 \cdot v_2 = 1 \cdot 4 = 4,$

$v_4 = u_4 \cdot v_3 = 3 \cdot 4 = 2,$

$v_5 = u_5 \cdot v_4 = 2 \cdot 2 = 0,$

$v_6 = u_6 \cdot v_5 = 3 \cdot 0 = 1,$

$v_7 = u_7 \cdot v_6 = 1 \cdot 1 = 0,$

$v_8 = u_8 \cdot v_7 = 1 \cdot 0 = 2,$

$v_9 = u_9 \cdot v_8 = 0 \cdot 2 = 2.$

Полученный зашифрованный текст $v = 244201022.$

Посмотрим, что получится после применения дешифрующего алгоритма к тексту v:

$u_1 = v_1/l = 2/2 = 0,$

$u_2 = v_2/v_1 = 4/2 = 3,$

$u_3 = v_3/v_2 = 4/4 = 1,$

$u_4 = v_4/v_3 = 2/4 = 3,$

$u_5 = v_5/v_4 = 0/2 = 2,$

$u_6 = v_6/v_5 = 1/0 = 3,$

$u_7 = v_7/v_6 = 0/1 = 1,$

$u_8 = v_8/v_7 = 2/0 = 1,$

$u_9 = v_9/v_8 = 2/2 = 0.$

Получен исходный текст $u = 031323110$.

Реализацию алгоритма Марковского для этого примера можно найти в Приложении 2: Программа А2.5 для шифрования и Программа А2.6 для дешифрования. С помощью этой программы можно построить реализацию алгоритма Марковского для любой правой бинарной квазигруппы. Отметим, что сложность программы увеличивается с ростом порядка используемой квазигруппы.

N -арные квазигруппы и их парастрофы могут быть также использованы для построения алгоритма Марковского [29, 166].

2.4. Обобщение алгоритма Марковского (Обобщенный Алгоритм 1)

Определим n -арную операцию f на множестве Q как совокупность $(n+1)$ -кортежей вида $\big(x_1, x_2, \dots, x_n, f(x_1, x_2, \dots, x_n)\big)$, где $x_1, x_2, \dots, x_n,\ f(x_1, x_2, \dots, x_n) \in Q$, это не что иное, как отображение множества Q^n в Q. При этом каждой последовательности $(x_1, x_2, \dots, x_n)$ поставлен в соответствие элемент $f(x_1, x_2, \dots, x_n)$.

Определение 2.4.1. *n-арным группоидом* (Q, f) называется непустое множество Q вместе с определенной на нем n-арной операцией f.

Определение 2.4.2. n-арный группоид (Q, f) называется обратимым на *i-м месте*, где $i = \overline{1,n}$, если уравнение: $f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n) = a_{n+1}$ однозначно разрешимо для любых элементов $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n, a_{n+1} \in Q$ [139].

В этом случае обратная операция ${}^{(i,n+1)}f(a_1, \dots, a_{i-1}, a_{n+1}, a_{i+1}, \dots, a_n) = x_i$ определяется единственным образом и для них выполнено условие:

$$\left.\begin{matrix} f\big(a_1, \dots, a_{i-1}, {}^{(i,n+1)}f(a_1, \dots, a_{i-1}, a_{n+1}, a_{i+1}, \dots, a_n), a_{i+1}, \dots, a_n\big) = a_{n+1} \\ {}^{(i,n+1)}f(a_1, \dots, a_{i-1}, f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n), a_{i+1}, \dots, a_n) = x_i \end{matrix}\right\}. \quad (\mathbf{2.1})$$

Определение 2.4.3. n-арный группоид (Q, f) с n-арной операцией f такой, что в уравнении $f(x_1, x_2, \dots, x_n) = x_{n+1}$ факт знания любых n элементов из множества $\{x_1, x_2, \dots, x_n, x_{n+1}\}$ позволяет однозначно определить оставшийся один элемент, называется *n -арной квазигруппой* [139, 29].

Определение 2.4.4. n-арный группоид (Q, f) называется n *-арной квазигруппой* , если на множестве Q существуют операции ${}^{(1,n+1)}f, {}^{(2,n+1)}f, \dots, {}^{(n,n+1)}f$ такие, что в алгебре $(Q, f, {}^{(1,n+1)}f, {}^{(2,n+1)}f, \dots, {}^{(n,n+1)}f)$ следующие тождества выполняются для всех $i = \overline{1,n}$:

$$\left.\begin{matrix} f\left(x_1, \dots, x_{i-1}, {}^{(i,n+1)}f(x_1, \dots, x_n), x_{i+1}, \dots, x_n\right) = x_i \\ {}^{(i,n+1)}f(x_1, \dots, x_{i-1}, f(x_1, \dots, x_n), x_{i+1}, \dots, x_n) = x_i \end{matrix}\right\}. \quad (\mathbf{2.2})$$

Рассмотрим модификацию криптоалгоритма Марковского, основанную на использовании n -арного группоида, обратимого на i -м месте. Криптоалгоритм, основанный на n-арной квазигруппе, обладает лучшими «смешивающими» свойствами, чем алгоритм основанный на бинарной квазигруппе [29, 167].

Бинарная (n-арная) квазигруппа, используемая в алгоритме Марковского, является его ключом. Число i-обратимых n-арных группоидов (число n фиксировано) больше, чем число n -арных квазигрупп (число n фиксировано). Этот факт послужил толчком к построению новых обобщений алгоритма Марковского.

Алгоритм 2.4.5. (Обобщенный Алгоритм 1). Пусть Q непустой конечный алфавит и $k -$ натуральное число, $u_j, v_j \in Q, j \in \{1, \dots k\}$.

Определим n-арный группоид (Q, f), который обратим на i-м месте, $i = \overline{1,n}$. Тогда группоид $(Q, {}^{(i,n+1)}f)$ будет определен однозначно. В этом случае имеет место равенство:

$${}^{(i,n+1)}f(v_1, \dots, v_{i-1}, v_n, v_i, \dots, v_{n-1}) =$$

$$= {}^{(i,n+1)}f(v_1, \dots, v_{i-1}, f(v_1, \dots, v_{i-1}, u_n, v_i, \dots, v_{n-1}), v_i, \dots, v_{n-1}) = u_n.$$

Выберем фиксированные элементы $l_1^{(n-1)^2}\left(l_1, l_2, \dots l_{(n-1)^2} \in Q\right)$, которые назовем лидерами.

Пусть $u_1, u_2, \dots u_k -$ это кортеж из k букв алфавита Q.

Предлагается следующая *процедура шифрования*:

$v_1 = f(l_1, l_2, \dots, l_{i-1}, u_1, l_i, \dots, l_{n-1})$,

$v_2 = f(l_n, l_{n+1}, \dots, l_{n+i-2}, u_2, l_{n+i-1}, \dots, l_{2n-2})$,…,

$v_{n-1} = f\left(l_{n^2-3n+3}, \dots, l_{n^2-3n+1+i}, u_{n-1}, l_{n^2-3n+2+i}, \dots, l_{(n-1)^2}\right)$,

$v_n = f(v_1, \dots, v_{i-1}, u_n, v_i, \dots, v_{n-1})$,

$v_{n+1} = f(v_2, \dots, v_i, u_{n+1}, v_{i+1}, \dots, v_n)$,

$v_{n+2} = f(v_3, \dots, v_{i+1}, u_{n+2}, v_{i+2}, \dots, v_{n+1})$,… .

Получаем следующий зашифрованный текст: $v_1, v_2, v_3, \dots, v_{n-1}, v_n, v_{n+1}, \dots$.

Алгоритм дешифрования строится аналогично бинарному случаю:

$u_1 = {}^{(i,n+1)}f(l_1, l_2, \dots, l_{i-1}, v_1, l_i, \dots, l_{n-1})$,

$u_2 = {}^{(i,n+1)}f(l_n, l_{n+1}, \dots, l_{n+i-2},\ v_2, l_{n+i-1}, \dots, l_{2n-2}),\dots,$

$u_{n-1} = {}^{(i,n+1)}f(\ l_{n^2-3n+3}, \dots,\ l_{n^2-3n+1+i}, v_{n-1}, l_{n^2-3n+2+i}, \dots, l_{(n-1)^2}),$

$u_n = {}^{(i,n+1)}f(v_1, \dots, v_{i-1}, v_n, v_i, \dots, v_{n-1}),$

$u_{n+1} = {}^{(i,n+1)}f(\ v_2, \dots, v_i, v_{n+1}, v_{i+1}, \dots, v_n),$

$u_{n+2} = {}^{(i,n+1)}f(v_3, \dots, v_{i+1}, v_{n+2}, v_{i+2}, \dots, v_{n+1}),\dots .$

Для обратимого на последнем месте группоида (частный случай, когда $i = n$) алгоритм шифрования имеет вид:

$v_1 = f(l_1, l_2, \dots, l_{n-1}, u_1),$

$v_2 = f(l_n, l_{n+1}, \dots, l_{2n-2},\ u_2),\ \dots,$

$v_{n-1} = f\left(l_{n^2-3n+3}, \dots, l_{(n-1)^2}, u_{n-1}\right),$

$v_n = f(v_1, \dots, v_{n-1}, u_n),$

$v_{n+1} = f(v_2, \dots, v_n, u_{n+1}),$

$v_{n+2} = f(v_3, \dots, v_{n+1}, u_{n+2}),\ \dots .$

Алгоритм дешифрования в этом случае принимает вид:

$u_1 = {}^{(n,n+1)}f(l_1, l_2, \dots, l_{n-1},\ v_1),$

$u_2 = {}^{(n,n+1)}f(l_n, l_{n+1}, \dots, l_{2n-2},\ v_2),\dots,$

$u_{n-1} = {}^{(n,n+1)}f(\ l_{n^2-3n+3}, \dots, l_{(n-1)^2}, v_{n-1}),$

$u_n = {}^{(n,n+1)}f(v_1, \dots, v_{n-1},\ v_n),$

$u_{n+1} = {}^{(n,n+1)}f(\ v_2, \dots, v_n,\ v_{n+1}),$

$u_{n+2} = {}^{(n,n+1)}f(v_3, \dots, v_{n+1},\ v_{n+2}),\dots,$

и тогда имеет место равенство:

$${}^{(n,n+1)}f(v_1, \dots, v_{n-1},\ v_n) = {}^{(n,n+1)}f(v_1, \dots, v_{n-1},\ f(v_1, \dots, v_{n-1}, u_n)) = u_n.$$

Пример 2.4.6. Возьмем тернарный группоид $(R_3, f),\ R_3 = \{0,1,2\}$, который определен над кольцом классов вычетов по модулю $3 - (R_3, +,\ \cdot)$ и обратим на третьем месте. Тернарная операция f на множестве R_3 определена следующим образом:

$f(x_1, x_2, x_3) = \alpha x_1 + \beta x_2 + \gamma x_3 = x_4$, где

$\alpha 0 = 1,\ \ \alpha 1 = 1,\ \ \alpha 2 = 0,$

$\beta 0 = 1,\ \ \beta 1 = 1,\ \ \beta 2 = 2,$

$\gamma 0 = 1,\ \ \gamma 1 = 2,\ \ \gamma 2 = 0.$

Таблица 2.8. Значения функции шифрования f

№	Значение	№	Значение	№	Значение
(1)	$f(0,0,0)=0$	(10)	$f(1,0,0)=0$	(19)	$f(2,0,0)=2$
(2)	$f(0,0,1)=1$	(11)	$f(1,0,1)=1$	(20)	$f(2,0,1)=0$
(3)	$f(0,0,2)=2$	(12)	$f(1,0,2)=2$	(21)	$f(2,0,2)=1$
(4)	$f(0,1,0)=0$	(13)	$f(1,1,0)=0$	(22)	$f(2,1,0)=2$
(5)	$f(0,1,1)=1$	(14)	$f(1,1,1)=1$	(23)	$f(2,1,1)=0$
(6)	$f(0,1,2)=2$	(15)	$f(1,1,2)=2$	(24)	$f(2,1,2)=1$
(7)	$f(0,2,0)=1$	(16)	$f(1,2,0)=1$	(25)	$f(2,2,0)=0$
(8)	$f(0,2,1)=2$	(17)	$f(1,2,1)=2$	(26)	$f(2,2,1)=1$
(9)	$f(0,2,2)=0$	(18)	$f(1,2,2)=0$	(27)	$f(2,2,2)=2$

Обратной операцией для f будет следующая операция:

${}^{(3,4)}f(x_1,x_2,x_4)=x_3=\gamma^{-1}(2\cdot\alpha x_1+2\cdot\beta x_2+x_4)$, где

$\gamma^{-1}(0)=2,\ \ \gamma^{-1}(1)=0,\ \ \gamma^{-1}(2)=1$.

Проверим, что операции f и ${}^{(3,4)}f$ взаимно обратны друг другу:

$f(x_1,x_2,x_3)=f\left(x_1,x_2,{}^{(3,4)}f(x_1,x_2,x_4)\right)=$

$=\alpha x_1+\beta\ x_2+\gamma\gamma^{-1}(2\cdot\alpha x_1+2\cdot\beta x_2+x_4)=\alpha x_1+\beta\ x_2+2\cdot\alpha x_1+2\cdot\beta x_2+$

$+x_4=\ x_4;$

${}^{(3,4)}f(x_1,x_2,x_4)={}^{(3,4)}f\left(x_1,x_2,f(x_1,x_2,x_3)\right)=\gamma^{-1}(2\cdot\alpha x_1+2\cdot\beta x_2+x_1+\beta\ x_2+$

$+\gamma x_3)=\gamma^{-1}(\gamma x_3)=x_3.$

Таблица 2.9. Значения функции дешифрования ${}^{(3,4)}f$

№	Значение	№	Значение	№	Значение
(1)	${}^{(3,4)}f(0,0,0)=0$	(10)	${}^{(3,4)}f(1,0,0)=0$	(19)	${}^{(3,4)}f(2,0,0)=1$
(2)	${}^{(3,4)}f(0,0,1)=1$	(11)	${}^{(3,4)}f(1,0,1)=1$	(20)	${}^{(3,4)}f(2,0,1)=2$
(3)	${}^{(3,4)}f(0,0,2)=2$	(12)	${}^{(3,4)}f(1,0,2)=2$	(21)	${}^{(3,4)}f(2,0,2)=0$
(4)	${}^{(3,4)}f(0,1,0)=0$	(13)	${}^{(3,4)}f(1,1,0)=0$	(22)	${}^{(3,4)}f(2,1,0)=1$
(5)	${}^{(3,4)}f(0,1,1)=1$	(14)	${}^{(3,4)}f(1,1,1)=1$	(23)	${}^{(3,4)}f(2,1,1)=2$
(6)	${}^{(3,4)}f(0,1,2)=2$	(15)	${}^{(3,4)}f(1,1,2)=2$	(24)	${}^{(3,4)}f(2,1,2)=0$
(7)	${}^{(3,4)}f(0,2,0)=2$	(16)	${}^{(3,4)}f(1,2,0)=2$	(25)	${}^{(3,4)}f(2,2,0)=0$
(8)	${}^{(3,4)}f(0,2,1)=0$	(17)	${}^{(3,4)}f(1,2,1)=0$	(26)	${}^{(3,4)}f(2,2,1)=1$
(9)	${}^{(3,4)}f(0,2,2)=1$	(18)	${}^{(3,4)}f(1,2,2)=1$	(27)	${}^{(3,4)}f(2,2,2)=2$

Элементы $l_1=1,l_2=2,l_3=0,l_4=1$ выберем в качестве лидеров.

Открытый текст 021022110 будет преобразован в следующий криптотекст:

$f(l_1, l_2, u_1) = f(1,2,0) = 1 = v_1,$

$f(l_3, l_4, u_2) = f(0,1,2) = 2 = v_2,$

$f(v_1, v_2, u_3) = f(1,2,1) = 2 = v_3,$

$f(v_2, v_3, u_4) = f(2,2,0) = 0 = v_4,$

$f(v_3, v_4, u_5) = f(2,0,2) = 1 = v_5,$

$f(v_4, v_5, u_6) = f(0,1,2) = 2 = v_6,$

$f(v_5, v_6, u_7) = f(1,2,1) = 2 = v_7,$

$f(v_6, v_7, u_8) = f(2,2,1) = 1 = v_8,$

$f(v_7, v_8, u_9) = f(2,1,0) = 2 = v_9.$

Примененим функцию дешифрования к $v = 122012212$:

${}^{(3,4)}f(l_1, l_2, v_1) = {}^{(3,4)}f(1,2,1) = 0,$

${}^{(3,4)}f(l_3, l_4, v_2) = {}^{(3,4)}f(0,1,2) = 2,$

${}^{(3,4)}f(v_1, v_2, v_3) = {}^{(3,4)}f(1,2,2) = 1,$

${}^{(3,4)}f(v_2, v_3, v_4) = {}^{(3,4)}f(2,2,0) = 0,$

${}^{(3,4)}f(v_3, v_4, v_5) = {}^{(3,4)}f(2,0,1) = 2,$

${}^{(3,4)}f(v_4, v_5, v_6) = {}^{(3,4)}f(0,1,2) = 2,$

${}^{(3,4)}f(v_5, v_6, v_7) = {}^{(3,4)}f(1,2,2) = 1,$

${}^{(3,4)}f(v_6, v_7, v_8) = {}^{(3,4)}f(2,2,1) = 1,$

${}^{(3,4)}f(v_7, v_8, v_9) = {}^{(3,4)}f(2,1,2) = 0.$

$u = 021022110$ – это и есть исходный текст.

Программная реализация шифрования и дешифрования с использованием Обобщенного Алгоритма 1 для этого примера приведена в Приложении 2: Программа А2.7 для шифрования и Программа А2.8 для дешифрования. Важно учитывать, что длина текста задается в начале программы, затем вводятся значения лидеров. Таблица шифрования или дешифрования вводится для каждого примера индивидуально, после чего происходит обработка открытого текста или зашифрованного текста.

Пример 2.4.7. Возьмем тернарный группоид (R_3, f), $R_3 = \{0,1,2\}$, который определен над кольцом классов вычетов по модулю 3 $-$ $(R_3, +, \cdot)$ и, который обратим на первом месте. Операция f на R_3 определяется так: $f(x_1, x_2, x_3) = \alpha x_1 + \beta x_2 + \gamma x_3 = x_4$, где

$\alpha 0 = 2, \ \ \alpha 1 = 0, \ \ \alpha 2 = 1,$

$\beta 0 = 0, \ \ \beta 1 = 1, \ \ \beta 2 = 1,$

$\gamma 0 = 2,\ \gamma 1 = 0,\ \gamma 2 = 0.$

Обратная операция для $f -$ это ${}^{(1,4)}f$ или (14)-парастроф вида:

${}^{(1,4)}f(x_4, x_2, x_3) = x_1 = \alpha^{-1}(x_4 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3),$ где

$\alpha^{-1}(0) = 1,\ \alpha^{-1}(1) = 2,\ \alpha^{-1}(2) = 0\,.$

Сделаем проверку: $f(x_1, x_2, x_3) = f\big({}^{(1,4)}f(x_4, x_2, x_3), x_2, x_3\big) =$

$= \alpha(\alpha^{-1}(x_4 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3) + \beta x_2 + \gamma x_3 =$

$= x_4 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3 + \beta\, x_2 + \gamma x_3 = x_4,$

${}^{(1,4)}f(x_4, x_2, x_3) = {}^{(1,4)}f(f(x_1, x_2, x_3), x_2, x_3) =$

$= \alpha^{-1}(\alpha x_1 + \beta\, x_2 + \gamma x_3 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3) = \alpha^{-1}(\alpha x_1) = x_1.$

Таблицы значений функций шифрования и дешифрования можно найти в Приложении 3 (Таблица А3.1 и Таблица А3.2 соответственно).

В качестве лидеров возьмем элементы: $l_1 = 0, l_2 = 2, l_3 = 1, l_4 = 2.$

Тогда открытый текст 021022110 будет преобразован по следующему алгоритму:

$f(u_1, l_1, l_2) = f(0,0,2) = 2 = v_1,$

$f(u_2, l_3, l_4) = f(2,1,2) = 2 = v_2,$

$f(u_3, v_1, v_2) = f(1,2,2) = 1 = v_3,$

$f(u_4, v_2, v_3) = f(0,2,1) = 0 = v_4,$

$f(u_5, v_3, v_4) = f(2,1,0) = 1 = v_5,$

$f(u_6, v_4, v_5) = f(2,0,1) = 1 = v_6,$

$f(u_7, v_5, v_6) = f(1,1,1) = 1 = v_7,$

$f(u_8, v_6, v_7) = f(1,1,1) = 1 = v_8,$

$f(u_9, v_7, v_8) = f(0,1,1) = 0 = v_9.$

Проверим работу алгоритма применив процедуру дешифрования к тексту $v = 221011110$:

${}^{(1,4)}f(v_1, l_1, l_2) = {}^{(1,4)}f(2,0,2) = 0 = u_1,$

${}^{(1,4)}f(v_2, l_3, l_4) = {}^{(1,4)}f(2,1,2) = 2 = u_2,$

${}^{(1,4)}f(v_3, v_1, v_2) = {}^{(1,4)}f(1,2,2) = 1 = u_3,$

${}^{(1,4)}f(v_4, v_2, v_3) = {}^{(1,4)}f(0,2,1) = 0 = u_4,$

${}^{(1,4)}f(v_5, v_3, v_4) = {}^{(1,4)}f(1,1,0) = 2 = u_5,$

${}^{(1,4)}f(v_6, v_4, v_5) = {}^{(1,4)}f(1,0,1) = 2 = u_6,$

${}^{(1,4)}f(v_7, v_5, v_6) = {}^{(1,4)}f(1,1,1) = 1 = u_7,$

${}^{(1,4)}f(v_8, v_6, v_7) = {}^{(1,4)}f(1,1,1) = 1 = u_8,$

$^{(1,4)}f(v_9, v_7, v_8) = {}^{(1,4)}f(0,1,1) = 0 = u_9$.

Получен исходный открытый текст, а значит алгоритм работает корректно.

Сложность алгоритма возрастает с ростом арности используемой операции.

Пример 2.4.8. Теперь возьмем 4-арный группоид (R_3, f), $R_3 = \{0,1,2\}$, который определен над кольцом классов вычетов по модулю 3 $-$ $(R_3, +, \cdot)$ и обратим на втором месте. 4-арная операция f на множестве R_3 определена так:

$f(x_1, x_2, x_3, x_4) = \alpha x_1 + \beta x_2 + \gamma x_3 + \delta x_4 = x_5$, где

$\alpha 0 = 2, \quad \alpha 1 = 0, \quad \alpha 2 = 1,$

$\beta 0 = 1, \quad \beta 1 = 2, \quad \beta 2 = 0,$

$\gamma 0 = 2, \quad \gamma 1 = 1, \quad \gamma 2 = 0,$

$\delta 0 = 0, \quad \delta 1 = 1, \quad \delta 2 = 2.$

Найдем значения операции f для всех допустимых наборов, которых будет 81:

Таблица 2.10. Значения функции шифрования f

№	Значение	№	Значение	№	Значение
(1)	$f(0,0,0,0) = 2$	(28)	$f(1,0,0,0) = 0$	(55)	$f(2,0,0,0) = 1$
(2)	$f(0,0,0,1) = 0$	(29)	$f(1,0,0,1) = 1$	(56)	$f(2,0,0,1) = 2$
(3)	$f(0,0,0,2) = 1$	(30)	$f(1,0,0,2) = 2$	(57)	$f(2,0,0,2) = 0$
(4)	$f(0,0,1,0) = 1$	(31)	$f(1,0,1,0) = 2$	(58)	$f(2,0,1,0) = 0$
(5)	$f(0,0,1,1) = 2$	(32)	$f(1,0,1,1) = 0$	(59)	$f(2,0,1,1) = 1$
(6)	$f(0,0,1,2) = 0$	(33)	$f(1,0,1,2) = 1$	(60)	$f(2,0,1,2) = 2$
(7)	$f(0,0,2,0) = 0$	(34)	$f(1,0,2,0) = 1$	(61)	$f(2,0,2,0) = 2$
(8)	$f(0,0,2,1) = 1$	(35)	$f(1,0,2,1) = 2$	(62)	$f(2,0,2,1) = 0$
(9)	$f(0,0,2,2) = 2$	(36)	$f(1,0,2,2) = 0$	(63)	$f(2,0,2,2) = 1$
(10)	$f(0,1,0,0) = 0$	(37)	$f(1,1,0,0) = 1$	(64)	$f(2,1,0,0) = 2$
(11)	$f(0,1,0,1) = 1$	(38)	$f(1,1,0,1) = 2$	(65)	$f(2,1,0,1) = 0$
(12)	$f(0,1,0,2) = 2$	(39)	$f(1,1,0,2) = 0$	(66)	$f(2,1,0,2) = 1$
(13)	$f(0,1,1,0) = 2$	(40)	$f(1,1,1,0) = 0$	(67)	$f(2,1,1,0) = 1$
(14)	$f(0,1,1,1) = 0$	(41)	$f(1,1,1,1) = 1$	(68)	$f(2,1,1,1) = 2$
(15)	$f(0,1,1,2) = 1$	(42)	$f(1,1,1,2) = 2$	(69)	$f(2,1,1,2) = 0$
(16)	$f(0,1,2,0) = 1$	(43)	$f(1,1,2,0) = 2$	(70)	$f(2,1,2,0) = 0$
(17)	$f(0,1,2,1) = 2$	(44)	$f(1,1,2,1) = 0$	(71)	$f(2,1,2,1) = 1$
(18)	$f(0,1,2,2) = 0$	(45)	$f(1,1,2,2) = 1$	(72)	$f(2,1,2,2) = 2$
(19)	$f(0,2,0,0) = 1$	(46)	$f(1,2,0,0) = 2$	(73)	$f(2,2,0,0) = 0$

(20)	$f(0,2,0,1) = 2$	(47)	$f(1,2,0,1) = 0$	(74)	$f(2,2,0,1) = 1$
(21)	$f(0,2,0,2) = 0$	(48)	$f(1,2,0,2) = 1$	(75)	$f(2,2,0,2) = 2$
(22)	$f(0,2,1,0) = 0$	(49)	$f(1,2,1,0) = 1$	(76)	$f(2,2,1,0) = 2$
(23)	$f(0,2,1,1) = 1$	(50)	$f(1,2,1,1) = 2$	(77)	$f(2,2,1,1) = 0$
(24)	$f(0,2,1,2) = 2$	(51)	$f(1,2,1,2) = 0$	(78)	$f(2,2,1,2) = 1$
(25)	$f(0,2,2,0) = 2$	(52)	$f(1,2,2,0) = 0$	(79)	$f(2,2,2,0) = 1$
(26)	$f(0,2,2,1) = 0$	(53)	$f(1,2,2,1) = 1$	(80)	$f(2,2,2,1) = 2$
(27)	$f(0,2,2,2) = 1$	(54)	$f(1,2,2,2) = 2$	(81)	$f(2,2,2,2) = 0$

В этом случае: ${}^{(2,5)}f(x_1, x_5, x_3, x_4) = x_2 = \beta^{-1}(2 \cdot \alpha x_1 + x_5 + 2 \cdot \gamma x_3 + 2 \cdot \delta x_4)$, где

$\beta^{-1}\,0 = 2, \qquad \beta^{-1}1 = 0, \qquad \beta^{-1}2 = 1.$

Проверка: $f(x_1, x_2, x_3, x_4) = f\big(x_1, {}^{(2,5)}f(x_1, x_5, x_3, x_4), x_3, x_4\big) =$

$= \alpha x_1 + \beta\, \beta^{-1}(2 \cdot \alpha x_1 + x_5 + 2 \cdot \gamma x_3 + 2 \cdot \delta x_4) + \gamma x_3 + \delta x_4 =$

$= \alpha x_1 + 2 \cdot \alpha x_1 + x_5 + 2 \cdot \gamma x_3 + 2 \cdot \delta x_4 + \gamma x_3 + \delta x_4 = x_5,$

${}^{(2,5)}f(x_1, x_5, x_3, x_4) = {}^{(2,5)}f(x_1, f(x_1, x_2, x_3, x_4), x_3, x_4) =$

$= \beta^{-1}(2 \cdot \alpha x_1 + \alpha x_1 + \beta\, x_2 + \gamma x_3 + \delta x_4 + 2 \cdot \gamma x_3 + 2 \cdot \delta x_4) = \beta^{-1}\beta\, x_2 = x_2.$

Таблица 2.11. Значения функции дешифрования ${}^{(2,5)}f$

№	Значение	№	Значение	№	Значение
(1)	${}^{(2,5)}f(0,0,0,0) = 1$	(28)	${}^{(2,5)}f(1,0,0,0) = 0$	(55)	${}^{(2,5)}f(2,0,0,0) = 2$
(2)	${}^{(2,5)}f(0,0,0,1) = 0$	(29)	${}^{(2,5)}f(1,0,0,1) = 2$	(56)	${}^{(2,5)}f(2,0,0,1) = 1$
(3)	${}^{(2,5)}f(0,0,0,2) = 2$	(30)	${}^{(2,5)}f(1,0,0,2) = 1$	(57)	${}^{(2,5)}f(2,0,0,2) = 0$
(4)	${}^{(2,5)}f(0,0,1,0) = 2$	(31)	${}^{(2,5)}f(1,0,1,0) = 1$	(58)	${}^{(2,5)}f(2,0,1,0) = 0$
(5)	${}^{(2,5)}f(0,0,1,1) = 1$	(32)	${}^{(2,5)}f(1,0,1,1) = 0$	(59)	${}^{(2,5)}f(2,0,1,1) = 2$
(6)	${}^{(2,5)}f(0,0,1,2) = 0$	(33)	${}^{(2,5)}f(1,0,1,2) = 2$	(60)	${}^{(2,5)}f(2,0,1,2) = 1$
(7)	${}^{(2,5)}f(0,0,2,0) = 0$	(34)	${}^{(2,5)}f(1,0,2,0) = 2$	(61)	${}^{(2,5)}f(2,0,2,0) = 1$
(8)	${}^{(2,5)}f(0,0,2,1) = 2$	(35)	${}^{(2,5)}f(1,0,2,1) = 1$	(62)	${}^{(2,5)}f(2,0,2,1) = 0$
(9)	${}^{(2,5)}f(0,0,2,2) = 1$	(36)	${}^{(2,5)}f(1,0,2,2) = 0$	(63)	${}^{(2,5)}f(2,0,2,2) = 2$
(10)	${}^{(2,5)}f(0,1,0,0) = 2$	(37)	${}^{(2,5)}f(1,1,0,0) = 1$	(64)	${}^{(2,5)}f(2,1,0,0) = 0$
(11)	${}^{(2,5)}f(0,1,0,1) = 1$	(38)	${}^{(2,5)}f(1,1,0,1) = 0$	(65)	${}^{(2,5)}f(2,1,0,1) = 2$
(12)	${}^{(2,5)}f(0,1,0,2) = 0$	(39)	${}^{(2,5)}f(1,1,0,2) = 2$	(66)	${}^{(2,5)}f(2,1,0,2) = 1$
(13)	${}^{(2,5)}f(0,1,1,0) = 0$	(40)	${}^{(2,5)}f(1,1,1,0) = 2$	(67)	${}^{(2,5)}f(2,1,1,0) = 1$
(14)	${}^{(2,5)}f(0,1,1,1) = 2$	(41)	${}^{(2,5)}f(1,1,1,1) = 1$	(68)	${}^{(2,5)}f(2,1,1,1) = 0$
(15)	${}^{(2,5)}f(0,1,1,2) = 1$	(42)	${}^{(2,5)}f(1,1,1,2) = 0$	(69)	${}^{(2,5)}f(2,1,1,2) = 2$

(16)	${}^{(2,5)}f(0,1,2,0) = 1$	(43)	${}^{(2,5)}f(1,1,2,0) = 0$	(70)	${}^{(2,5)}f(2,1,2,0) = 2$
(17)	${}^{(2,5)}f(0,1,2,1) = 0$	(44)	${}^{(2,5)}f(1,1,2,1) = 2$	(71)	${}^{(2,5)}f(2,1,2,1) = 1$
(18)	${}^{(2,5)}f(0,1,2,2) = 2$	(45)	${}^{(2,5)}f(1,1,2,2) = 1$	(72)	${}^{(2,5)}f(2,1,2,2) = 0$
(19)	${}^{(2,5)}f(0,2,0,0) = 0$	(46)	${}^{(2,5)}f(1,2,0,0) = 2$	(73)	${}^{(2,5)}f(2,2,0,0) = 1$
(20)	${}^{(2,5)}f(0,2,0,1) = 2$	(47)	${}^{(2,5)}f(1,2,0,1) = 1$	(74)	${}^{(2,5)}f(2,2,0,1) = 0$
(21)	${}^{(2,5)}f(0,2,0,2) = 1$	(48)	${}^{(2,5)}f(1,2,0,2) = 0$	(75)	${}^{(2,5)}f(2,2,0,2) = 2$
(22)	${}^{(2,5)}f(0,2,1,0) = 1$	(49)	${}^{(2,5)}f(1,2,1,0) = 0$	(76)	${}^{(2,5)}f(2,2,1,0) = 2$
(23)	${}^{(2,5)}f(0,2,1,1) = 0$	(50)	${}^{(2,5)}f(1,2,1,1) = 2$	(77)	${}^{(2,5)}f(2,2,1,1) = 1$
(24)	${}^{(2,5)}f(0,2,1,2) = 2$	(51)	${}^{(2,5)}f(1,2,1,2) = 1$	(78)	${}^{(2,5)}f(2,2,1,2) = 0$
(25)	${}^{(2,5)}f(0,2,2,0) = 2$	(52)	${}^{(2,5)}f(1,2,2,0) = 1$	(79)	${}^{(2,5)}f(2,2,2,0) = 0$
(26)	${}^{(2,5)}f(0,2,2,1) = 1$	(53)	${}^{(2,5)}f(1,2,2,1) = 0$	(80)	${}^{(2,5)}f(2,2,2,1) = 2$
(27)	${}^{(2,5)}f(0,2,2,2) = 0$	(54)	${}^{(2,5)}f(1,2,2,2) = 2$	(81)	${}^{(2,5)}f(2,2,2,2) = 1$

Используем в качестве лидеров следующие 9 элементов:

$$l_1 = 0,\ l_2 = 2,\ l_3 = 1,\ l_4 = 2,\ l_5 = 0,\ l_6 = 2,\ l_7 = 1,\ l_8 = 2,\ l_9 = 0.$$

Для открытого текста 201121001 процесс шифрования имеет вид:

$f(l_1, u_1, l_2, l_3) = f(0,2,2,1) = 0 = v_1,$

$f(l_4, u_2, l_5, l_6) = f(2,0,0,2) = 0 = v_2,$

$f(l_7, u_3, l_8, l_9) = f(1,1,2,0) = 2 = v_3,$

$f(v_1, u_4, v_2, v_3) = f(0,1,0,2) = 2 = v_4,$

$f(v_2, u_5, v_3, v_4) = f(0,2,2,2) = 1 = v_5,$

$f(v_3, u_6, v_4, v_5) = f(2,1,2,1) = 1 = v_6,$

$f(v_4, u_7, v_5, v_6) = f(2,0,1,1) = 1 = v_7,$

$f(v_5, u_8, v_6, v_7) = f(1,0,1,1) = 0 = v_8,$

$f(v_6, u_9, v_7, v_8) = f(1,1,1,0) = 0 = v_9.$

Применив дешифрующую процедуру к тексту $v = 002211100$ получим:

${}^{(2,5)}f(l_1, v_1, l_2, l_3) = {}^{(2,5)}f(0,0,2,1) = 2 = u_1,$

${}^{(2,5)}f(l_4, v_2, l_5, l_6) = {}^{(2,5)}f(2,0,0,2) = 0 = u_2,$

${}^{(2,5)}f(l_7, v_3, l_8, l_9) = {}^{(2,5)}f(1,2,2,0) = 1 = u_3,$

${}^{(2,5)}f(v_1, v_4, v_2, v_3) = {}^{(2,5)}f(0,2,0,2) = 1 = u_4,$

${}^{(2,5)}f(v_2, v_5, v_3, v_4) = {}^{(2,5)}f(0,1,2,2) = 2 = u_5,$

${}^{(2,5)}f(v_3, v_6, v_4, v_5) = {}^{(2,5)}f(2,1,2,1) = 1 = u_6,$

$^{(2,5)}f(v_4, v_7, v_5, v_6) = {}^{(2,5)}f(2,1,1,1) = 0 = u_7,$

$^{(2,5)}f(v_5, v_8, v_6, v_7) = {}^{(2,5)}f(1,0,1,1) = 0 = u_8,$

$^{(2,5)}f(v_6, v_9, v_7, v_8) = {}^{(2,5)}f(1,0,1,0) = 1 = u_9.$

Алгоритм работает корректно.

Замечание 2.4.9. Важным условием правильной работы алгоритма является однозначное указание подстановки на месте обратимости операции.

Полученный Обобщенный Алгоритм 1 в свою очередь можно модифицировать. Для этого воспользуемся таким понятием, как трансляция.

2.5. Обобщение алгоритма Марковского (Обобщенный Алгоритм 2)

Трансляция i-обратимого n-арного группоида (Q, f) $(n > 2)$ обозначается так: $T(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n)$, где $a_i \in Q$ для всех $i = \overline{1,n}$ и $T(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n)x = f(a_1, \dots, a_{i-1}, x, a_{i+1}, \dots, a_n)$ для всех $x \in Q$.

Из определения i-обратимого n-арного группоида (Q, f) следует, что любая трансляция n-арного группоида (Q, f) это некоторая перестановка элементов из Q. Следующая ниже лемма верна для любого значения индекса i.

Лемма 2.5.1. Если ${}_fT(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n)$ трансляция i-обратимого n-арного группоида (Q, f), тогда:

${}_fT^{-1}(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n) = {}_{(i,n+1)_f}T(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n).$

Доказательство. Для удобства в доказательстве опускается символ f в обозначении трансляции группоида (Q, f). Воспользовавшись (2.2) получим:

$T^{-1}(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n)(T(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n)x) =$

$= T^{-1}(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n)f(a_1, \dots, a_{i-1}, x, a_{i+1}, \dots, a_n) =$

$= {}^{(i,n+1)}f(a_1, \dots, a_{i-1}, f(a_1, \dots, a_{i-1}, x, a_{i+1}, \dots, a_n), a_{i+1}, \dots, a_n) = x.$

Если в Лемме 2.5.1. положить $i = n$, тогда получим:

${}_fT^{-1}(a_1, \dots, a_{n-1}, -) = {}_{(n,n+1)_f}T(a_1, \dots, a_{n-1}, -).$

Алгоритм 2.5.2. (Обобщенный Алгоритм 2). Пусть Q конечный непустой алфавит и k – натуральное число, $u_j, v_j \in Q, j \in \{1, \dots k\}$. Определим n-арный группоид (Q, f), который будет обратим на i-м месте. Ясно, что группоид $(Q, {}^{(i,n+1)}f)$ определится однозначным образом.

Выберем лидеры: $l_1^{(n^2-n)/2}$ $\left(l_1, l_2, \dots l_{(n^2-n)/2} \in Q\right)$.

Возьмем $u_1, u_2, \dots u_k -$ кортеж k букв из алфавита Q и натуральные числа: $a, b, c, d \dots$, которые будут служить степенями используемых в алгоритме трансляций.

Получим алгоритм шифрования вида:

$v_1 = T^a(l_1, l_2, \dots, l_{i-2}, l_{i-1}, u_1^a, l_i, \dots, l_{n-1}),$

$v_2 = T^b(l_n, l_{n+1}, \dots, l_{n+i-3}, v_1, u_2^b, l_{n+i-2}, \dots, l_{2n-3}),$

$v_3 = T^c(l_{2n-2}, l_{2n-1}, \dots, l_{2n-i+4}, v_1, v_2, u_3^c, l_{2n-i+3}, \dots, l_{3n-6}), \dots,$

$v_{n-1} = T^d(l_{(n^2-n)/2}, v_1, \dots, v_{i-2}, u_{n-1}^d, v_{i-1}, \dots, v_{n-2}),$

$v_n = T^e(v_1, v_2, \dots, v_{i-1}, u_n^e, v_i, \dots, v_{n-1}),$

$v_{n+1} = T^f(v_2, v_3, \dots, v_i, u_{n+1}^f, v_{i+1}, \dots, v_n), \dots .$

В алгоритме использовались следующие обозначения:

$u_1^a = \underbrace{f(f \dots f(l_1, l_2, \dots, l_{i-2}, l_{i-1}, u_1, l_i, \dots, l_{n-1}) \dots)}_{a \text{ раз}},$

$u_2^b = \underbrace{f(f \dots f(l_n, l_{n+1}, \dots, l_{n+i-3}, v_1, u_2, l_{n+i-2}, \dots, l_{2n-3}) \dots)}_{b \text{ раз}},$

$u_3^c = \underbrace{f(f \dots f(l_{2n-2}, l_{2n-1}, \dots, l_{2n-i+4}, v_1, v_2, u_3, l_{2n-i+3}, \dots, l_{3n-6}) \dots)}_{c \text{ раз}}, \dots,$

$u_n^e = \underbrace{f(f \dots f(v_1, v_2, \dots, v_{i-1}, u_n, v_i, \dots, v_{n-1}) \dots)}_{e \text{ раз}}, \dots .$

В результате получаем следующий зашифрованный текст: $v_1, v_2, \dots v_n, v_{n+1}, \dots$.

Для n-арного группоида, который обратим на последнем n-м месте, алгоритм шифрования примет вид:

$v_1 = T^a(l_1, l_2, \dots, l_{n-1}, u_1^a),$

$v_2 = T^b(l_n, l_{n+1}, \dots, l_{2n-3}, v_1, u_2^b), \dots,$

$v_{n-1} = T^d(l_{(n^2-n)/2}, v_1, \dots_{,n-2}, u_{n-1}^d),$

$v_n = T^e(v_1, v_2, \dots, v_{n-1}, u_n^e),$

$v_{n+1} = T^f(v_2, v_3, \dots, v_n, u_{n+1}^f), \dots .$

Учитывая Лемму 2.5.1 можно сказать, что алгоритм дешифрования для Обобщенного Алгоритма 2 может быть построен аналогично алгоритму дешифрования, данному для Обобщенного Алгоритма 1.

Например, для n-арного группоида, обратимого на n-м месте, алгоритм дешифрования будет иметь вид:

$u_1 = (T^a)^{-1}(l_1, l_2, \dots, l_{n-1}, v_1),$

$u_2 = (T^b)^{-1}(l_n, l_{n+1}, \dots, l_{2n-3}, v_1, v_2), \dots,$

$u_{n-1} = (T^c)^{-1}(l_{(n^2-n)/2}, v_1, \dots, v_{n-2}, v_{n-1})$,

$u_n = (T^d)^{-1}(v_1, v_2, \dots, v_{n-1}, v_n)$,

$u_{n+1} = (T^e)^{-1}(v_2, v_3, \dots, v_n, v_{n+1})$,

$u_{n+2} = (T^f)^{-1}(v_3, v_4, \dots, v_{n+1}, v_{n+2}), \dots$.

Работа и особенности Обобщенного Алгоритма 2 иллюстрируются приведенными ниже примерами.

Пример 2.5.3. Возьмем тернарный группоид (R_3, f), $R_3 = \{0,1,2\}$, который определен над кольцом классов вычетов по модулю 3$-$ $(R_3, +, \cdot)$. и обратим на последнем третьем месте. Определим тернарную операцию f на множестве R_3 так:

$f(x_1, x_2, x_3) = \alpha x_1 + \beta x_2 + \gamma x_3 = x_4$, где

$\alpha 0 = 2, \quad \alpha 1 = 0, \quad \alpha 2 = 1,$

$\beta 0 = 1, \quad \beta 1 = 0, \quad \beta 2 = 2,$

$\gamma 0 = 1, \quad \gamma 1 = 2, \quad \gamma 2 = 0.$

Ниже запись $T_{0,0}0 = 1$ будет обозначать, что $f(0,0,0) = 1$ и т.д. С учетом сказанного имеем:

Таблица 2.12. Значения функции шифрования f

№	Значение	№	Значение	№	Значение	№	Значение
(1)	$T_{0,0}0 = 1$	(8)	$T_{0,2}1 = 0$	(15)	$T_{1,1}2 = 0$	(22)	$T_{2,1}0 = 2$
(2)	$T_{0,0}1 = 2$	(9)	$T_{0,2}2 = 1$	(16)	$T_{1,2}0 = 0$	(23)	$T_{2,1}1 = 0$
(3)	$T_{0,0}2 = 0$	(10)	$T_{1,0}0 = 2$	(17)	$T_{1,2}1 = 1$	(24)	$T_{2,1}2 = 1$
(4)	$T_{0,1}0 = 0$	(11)	$T_{1,0}1 = 0$	(18)	$T_{1,2}2 = 2$	(25)	$T_{2,2}0 = 1$
(5)	$T_{0,1}1 = 1$	(12)	$T_{1,0}2 = 1$	(19)	$T_{2,0}0 = 0$	(26)	$T_{2,2}1 = 2$
(6)	$T_{0,1}2 = 2$	(13)	$T_{1,1}0 = 1$	(20)	$T_{2,0}1 = 1$	(27)	$T_{2,2}2 = 0$
(7)	$T_{0,2}0 = 2$	(14)	$T_{1,1}1 = 2$	(21)	$T_{2,0}2 = 2$		

Обратную операцию для f построим следующим образом:

${}^{(3,4)}f(x_1, x_2, x_4) = x_3 = \gamma^{-1}(2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + x_4)$, где

$\gamma^{-1}(0) = 2, \gamma^{-1}(1) = 0, \gamma^{-1}(2) = 1$.

Далее ${T^{-1}}_{0,0}0 = 2$ будет обозначать, что ${}^{(3,4)}f(0,0,0) = 2$ и т.д. и тогда имеем:

Таблица 2.13. Значения функции дешифрования ${}^{(3,4)}f$

№	Значение	№	Значение	№	Значение	№	Значение
(1)	${T^{-1}}_{0,0}0 = 2$	(8)	${T^{-1}}_{0,2}1 = 2$	(15)	${T^{-1}}_{1,1}2 = 1$	(22)	${T^{-1}}_{2,1}0 = 1$
(2)	${T^{-1}}_{0,0}1 = 0$	(9)	${T^{-1}}_{0,2}2 = 0$	(16)	${T^{-1}}_{1,2}0 = 0$	(23)	${T^{-1}}_{2,1}1 = 2$

(3)	${T^{-1}}_{0,0}2 = 1$	(10)	${T^{-1}}_{1,0}0 = 1$	(17)	${T^{-1}}_{1,2}1 = 1$	(24)	${T^{-1}}_{2,1}2 = 0$
(4)	${T^{-1}}_{0,1}0 = 0$	(11)	${T^{-1}}_{1,0}1 = 2$	(18)	${T^{-1}}_{1,2}2 = 2$	(25)	${T^{-1}}_{2,2}0 = 2$
(5)	${T^{-1}}_{0,1}1 = 1$	(12)	${T^{-1}}_{1,0}2 = 0$	(19)	${T^{-1}}_{2,0}0 = 0$	(26)	${T^{-1}}_{2,2}1 = 0$
(6)	${T^{-1}}_{0,1}2 = 2$	(13)	${T^{-1}}_{1,1}0 = 2$	(20)	${T^{-1}}_{2,0}1 = 1$	(27)	${T^{-1}}_{2,2}2 = 1$
(7)	${T^{-1}}_{0,2}0 = 1$	(14)	${T^{-1}}_{1,1}1 = 0$	(21)	${T^{-1}}_{2,0}2 = 2$		

Возьмем следующие элементы в качестве лидеров: $l_1 = 1,\ l_2 = 2,\ l_3 = 0$.

В Алгоритме 2.5.2. положим следующие степени трансляций: $a = 1, b = 2, c\ = 1,$ $d\ = 2$ и т.д.

Возьмем открытый текст 021022110 и подвергнем его шифрованию:

$v_1 = T^1_{l_1,l_2}u_1 = f(l_1, l_2, u_1) = f(1,2,0) = 0 \Rightarrow v_1 = 0,$

$v_2 = T^2_{l_3,v_1}u_2 = f\big(l_3, v_1, f(l_3, v_1, u_2)\big) = f\big(0,0, f(0,0,2)\big) = f(0,0,0)\ = 1\ \Rightarrow v_2 = 1,$

$v_3 = T^1_{v_1,v_2}u_3 = f(v_1, v_2, u_3) = f(0,1,1) = 1 \Rightarrow v_3 = 1,$

$v_4 = T^2_{v_2,v_3}u_4 = f\big(v_2, v_3, f(v_2, v_3, u_4)\big) = f\big(1,1, f(1,1,0)\big) = f(1,1,1)\ = 2 \Rightarrow v_4 = 2,$

$v_5 = T^1_{v_3,v_4}u_5 = f(v_3, v_4, u_5) = f(1,2,2) = 2 \Rightarrow v_5 = 2,$

$v_6 = T^2_{v_4,v_5}u_6 = f\big(v_4, v_5, f(v_4, v_5, u_6)\big) = f\big(2,2, f(2,2,2)\big) = f(2,2,0)\ = 1 \Rightarrow v_6 = 1,$

$v_7 = T^1_{v_5,v_6}u_7 = f(v_5, v_6, u_7) = f(2,1,1) = 0 \Rightarrow v_7 = 0,$

$v_8 = T^2_{v_6,v_7}u_8 = f\big(v_6, v_7, f(v_6, v_7, u_8)\big) = f\big(1,0, f(1,0,1)\big) = f(1,0,0)\ = 2 \Rightarrow v_8 = 2,$

$v_9 = T^1_{v_7,v_8}u_9 = f(v_7, v_8, u_9) = f(0,2,0) = 2 \Rightarrow v_9 = 2.$

В результате получим зашифрованный текст 011221022.

Заметим, что в условиях данного примера верен следующий факт: $T^{-1}(x, y, -) = T^2(x, y, -)$.

Теперь применим к шифротексту процедуру дешифрования.

$T^2_{l_1,l_2}v_1 = f(l_1, l_2, f(l_1, l_2, v_1)) = f\big(1,2, f(1,2,0)\big) = f(1,2,0) = {}^{(3,4)}f(1,2,0) = 0$

$\Rightarrow u_1 = 0,$

$T^1_{l_3,v_1}v_2 = f(l_3, v_1, v_2) = f(0,0,1) = 2\ \Rightarrow u_2 = 2,$

$T^2_{v_1,v_2}v_3 = f\big(v_1, v_2, f(v_1, v_2, v_3)\big) = f(0,1, f(0,1,1)) = f(0,1,1) = {}^{(3,4)}f(0,1,1) = 1$

$\Rightarrow u_3 = 1,$

$T^1_{v_2,v_3}v_4 = f(v_2, v_3, v_4\ ) = f(1,1,2)\ = 0 \Rightarrow u_4 = 0,$

$T^2_{v_3,v_4}v_5 = f\big(v_3, v_4, f(v_3, v_4, v_5)\big) = f(1,2, f(1,2,2)) = f(1,2,2) = {}^{(3,4)}f(1,2,2) = 2$

$\Rightarrow u_5 = 2,$

$T^1_{v_4,v_5}v_6 = f(v_4, v_5, v_6) = f(2,2,1) = 2 \Rightarrow u_6 = 2,$

$T^2_{v_5,v_6}v_7 = f\big(v_5, v_6, f(v_5, v_6, v_7)\big) = f(2,1,f(2,1,0)) = f(2,1,2) = {}^{(3,4)}f(2,1,0) = 1$

$\Rightarrow u_7 = 1,$

$T^1_{v_6v_7}v_8 = f(v_6, v_7, v_8) = f(1,0,2) = 1 \Rightarrow u_8 = 1,$

$T^2_{v_7,v_8}v_9 = f\big(v_7, v_8, f(v_7, v_8, v_9)\big) = f(0,2,f(0,2,2)) = f(0,2,1) = {}^{(3,4)}f(0,2,2) = 0$

$\Rightarrow u_9 = 0.$

Таким образом, получен исходный открытый текст 021022110.

Программная реализация шифрования и дешифрования с использованием Обобщенного Алгоритма 2 для этого примера приведена в Приложении 2: Программа А2. 9 для шифрования и Программа А2.10 для дешифрования.

Сравнивая характеристики программ для построенных обобщенных алгоритмов можно сделать вывод, что программы для второго алгоритма значительно сложнее и объемнее, что обусловлено использованием трансляций в программах второго типа.

Пример 2.5.4. Пусть тернарный группоид (R_3, f), $R_3 = \{0,1,2\}$, определен над кольцом $(R_3, +, \cdot)$ и обратим на первом месте. Операция f на R_3 определяется так:

$f(x_1, x_2, x_3) = \alpha x_1 + \beta x_2 + \gamma x_3 = x_4$, где

$\alpha 0 = 2,\ \alpha 1 = 0,\ \alpha 2 = 1,$

$\beta 0 = 2,\ \beta 1 = 0,\ \beta 2 = 1,$

$\gamma 0 = 1,\ \gamma 1 = 2,\ \gamma 2 = 0.$

Таблица 2.14. Значения функции шифрования *f*

№	Значение	№	Значение	№	Значение	№	Значение
(1)	$T_{0,0}0 = 2$	(8)	$T_{0,2}1 = 2$	(15)	$T_{1,1}2 = 0$	(22)	$T_{2,1}0 = 2$
(2)	$T_{0,0}1 = 0$	(9)	$T_{0,2}2 = 0$	(16)	$T_{1,2}0 = 2$	(23)	$T_{2,1}1 = 0$
(3)	$T_{0,0}2 = 1$	(10)	$T_{1,0}0 = 0$	(17)	$T_{1,2}1 = 0$	(24)	$T_{2,1}2 = 1$
(4)	$T_{0,1}0 = 0$	(11)	$T_{1,0}1 = 1$	(18)	$T_{1,2}2 = 1$	(25)	$T_{2,2}0 = 0$
(5)	$T_{0,1}1 = 1$	(12)	$T_{1,0}2 = 2$	(19)	$T_{2,0}0 = 1$	(26)	$T_{2,2}1 = 1$
(6)	$T_{0,1}2 = 2$	(13)	$T_{1,1}0 = 1$	(20)	$T_{2,0}1 = 2$	(27)	$T_{2,2}2 = 2$
(7)	$T_{0,2}0 = 1$	(14)	$T_{1,1}1 = 2$	(21)	$T_{2,0}2 = 0$		

Обратная операция для f имеет вид:

${}^{(1,4)}f(x_4, x_2, x_3) = x_1 = \alpha^{-1}(x_4 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3)$, где

$\alpha^{-1}(0) = 1,\ \alpha^{-1}(1) = 2,\ \alpha^{-1}(2) = 0\,.$

Лидерами служат элементы $l_1 = 1,\ l_2 = 2,\ l_3 = 0.$

В Алгоритме 2.5.2. чередуются степени: $a = 1,\ b = 2,\ c = 3,\ d = 1,\ e = 2,\ f = 3$ и т.д.

Возьмем открытый текст 021022110 и применим к нему алгоритм шифрования:

$v_1 = T^1_{u_1,l_1} l_2 = f(u_1, l_1, l_2) = f(0,1,2) = 2 \Rightarrow v_1 = 2,$

$v_2 = T^2_{u_2,l_3} v_1 = f(f(u_2, l_3, v_1), l_3, v_1) = f(f(2,0,2),0,2) = f(0,0,2) = 1 \Rightarrow v_2 = 1,$

$v_3 = T^3_{u_3,v_1} v_2 = f(f(f(u_3, v_1, v_2), v_1, v_2), v_1, v_2) = f(f(f(1,2,1), 2,1), 2,1) =$

$= f(f(0,2,1), 2,1) = f(2,2,1) = 1 \Rightarrow v_3 = 1,$

$v_4 = T^1_{u_4,v_2} v_3 = f(u_4, v_2, v_3) = f(0,1,1) = 1 \Rightarrow v_4 = 1,$

$v_5 = T^2_{u_5,v_3} v_4 = f(f(u_5, v_3, v_4), v_3, v_4) = f(f(2,1,1),1,1) = f(0,1,1) = 1 \Rightarrow v_5 = 1,$

$v_6 = T^3_{u_6,v_4} v_5 = f(f(f(u_6, v_4, v_5), v_4, v_5), v_4, v_5) = f(f(f(2,1,1), 1,1), 1,1) =$

$= f(f(0,1,1), 1,1) = f(1,1,1) = 2 \Rightarrow v_6 = 2,$

$v_7 = T^1_{u_7,v_5} v_6 = f(u_7, v_5, v_6) = f(1,1,2) = 0 \Rightarrow v_7 = 0,$

$v_8 = T^2_{u_8,v_6} v_7 = f(f(u_8, v_6, v_7), v_6, v_7) = f(f(1,2,0),2,0) = f(2,2,0) = 0 \Rightarrow v_8 = 0,$

$v_9 = T^3_{u_9,v_7} v_7 = f(f(f(u_9, v_7, v_8), v_7, v_8), v_7, v_8) = f(f(f(0,0,0), 0,0), 0,0) =$

$= f(f(2,0,0), 0,0) = f(1,0,0) = 0 \Rightarrow v_9 = 0.$

Получим зашифрованный текст 211112000.

Заметим, что в условиях данного примера верен следующий факт:

$T^{-1}(-, x, y) = T^2(-, x, y), T^{-2}(-, x, y) = T^1(-, x, y),\ T^{-3}(-, x, y) = T^3(-, x, y).$

Проведем процедуру дешифрования полученного зашифрованного текста:

$T^2_{v_1,l_1} l_2 = f(f(v_1, l_1, l_2), l_1, l_2) = f(f(2,1,2), 1,2) = f(1,1,2) = {}^{(1,4)}f(2,1,2) = 0 \Rightarrow$

$\Rightarrow u_1 = 0,$

$T^1_{v_2,l_3} v_1 = f(v_2, l_3, v_1) = f(1,0,2) = 2 \Rightarrow u_2 = 2,$

$T^3_{v_3,v_1} v_2 = f(f(f(v_3, v_1, v_2), v_1, v_2), v_1, v_2) = f(f(f(1,2,1),2,1), 2,1) =$

$= f(f(0,2,1),2,1) = f(2,2,1) = 1 \Rightarrow u_3 = 1,$

$T^2_{v_4,v_2} v_3 = f(f(v_4, v_2, v_3), v_2, v_3) = f(f(1,1,1), 1,1) = f(2,1,1) = {}^{(1,4)}f(1,1,1) = 0 \Rightarrow$

$\Rightarrow u_4 = 0,$

$T^1_{v_5,v_3} v_4 = f(v_5, v_3, v_4) = f(1,1,1) = 2 \Rightarrow u_5 = 2,$

$T^3_{v_6,v_4} v_5 = f(f(f(v_6, v_4, v_5), v_4, v_5), v_4, v_5) = f(f(f(2,1,1), 1,1), 1,1) =$

$= f(f(0,1,1), 1,1) = f(1,1,1) = 2 \Rightarrow u_6 = 2,$

$T^2_{v_7,v_5} v_6 = f(f(v_7, v_5, v_6), v_5, v_6) = f(f(0,1,2), 1,2) = f(2,1,2) = {}^{(1,4)}f(0,1,2) = 1 \Rightarrow$

$\Rightarrow u_7 = 1,$

$T^1_{v_8,v_6}v_7 = f(v_8, v_6, v_7) = f(0,2,0) = 1 \Rightarrow u_8 = 1,$

$T^3_{v_9,v_7}v_8 = f(f(f(v_9, v_7, v_8), v_7, v_8), v_7, v_8) = f(f(f(0,0,0), 0,0), 0,0) =$

$= f(f(2,0,0), 0,0) = f(1,0,0) = 0 \Rightarrow u_9 = 0.$

Получен исходный открытый текст 021022110.

Значения обратной функции ${}^{(1,4)}f$ легко находятся по таблице значений функции шифрования f т.е. из Таблицы 2.14.

Пример 2.5.5. Возьмем 4-арный группоид (R_3, f), $R_3 = \{0,1,2\}$, заданный над кольцом классов вычетов по модулю 3 $-(R_3, +, \cdot)$ и обратимый на четвертом месте. 4-арная операция f на множестве R_3 будет задаваться так:

$f(x_1, x_2, x_3, x_4) = \alpha x_1 + \beta x_2 + \gamma x_3 + \delta x_4 = x_5$, где

$\alpha 0 = 2, \quad \alpha 1 = 0, \quad \alpha 2 = 1,$

$\beta 0 = 1, \quad \beta 1 = 2, \quad \beta 2 = 0,$

$\gamma\, 0 = 2, \quad \gamma 1 = 1, \quad \gamma 2 = 0,$

$\delta 0 = 0, \quad \delta 1 = 1, \quad \delta 2 = 2.$

В этом случае: ${}^{(4,5)}f(x_1, x_2, x_3, x_5) = x_4 = \delta^{-1}(2 \cdot \alpha x_1 + 2 \cdot \beta\, x_2 + 2 \cdot \gamma x_3 + x_5)$, где

$\delta^{-1}\, 0 = 0, \quad \delta^{-1} 1 = 1, \quad \delta^{-1} 2 = 2.$

Таблицу значений функции шифрования можно найти в Приложении 3, Таблица А3.3.

Возьмем элементы $l_1 = 0,\ l_2 = 1,\ l_3 = 2,\ l_4 = 0,\ l_5 = 2,\ l_6 = 0$ в качестве лидеров.

В Обобщенном Алгоритме 2 положим $a = 1,\ b = 2,\ c = 1,\ d = 2$ и т.д.

Открытый текст 021022110 будет преобразован так:

$v_1 = T^1_{l_1,l_2,l_3}u_1 = f(l_1, l_2, l_3, u_1) = f(0,1,2,0) = 1 \Rightarrow v_1 = 1,$

$v_2 = T^2_{l_4,l_5,v_1}u_2 = f\big(l_4, l_5, v_1, f(l_4, l_5, v_1, u_2)\big) = f\big(0,2,1, f(0,2,1,2)\big) = f(0,2,1,2) = 2$

$\Rightarrow v_2 = 2,$

$v_3 = T^1_{l_6,v_1,v_2}u_3 = f(l_6, v_1, v_2, u_3) = f(0,1,2,1) = 2 \Rightarrow v_3 = 2,$

$v_4 = T^2_{v_1,v_2,v_3}u_4 = f\big(v_1, v_2, v_3, f(v_1, v_2, v_3, u_4)\big) = f\big(1,2,2, f(1,2,2,0)\big) = f(1,2,2,0) =$

$= 0 \Rightarrow v_4 = 0,$

$v_5 = T^1_{v_2,v_3,v_4}u_5 = f(v_2, v_3, v_4, u_5) = f(2,2,0,2) = 2 \Rightarrow v_5 = 2,$

$v_6 = T^2_{v_3,v_4,v_5}u_6 = f\big(v_3, v_4, v_5, f(v_3, v_4, v_5, u_6)\big) = f\big(2,0,2, f(2,0,2,2)\big) = f(2,0,2,1) =$

$= 0 \Rightarrow v_6 = 0,$

$v_7 = T^1_{v_4,v_5,v_6}u_7 = f(v_4, v_5, v_6, u_7) = f(0,2,0,1) = 2 \Rightarrow v_7 = 2,$

$v_8 = T^2_{v_5,v_6,v_7} u_8 = f\big(v_5, v_6, v_7, f(v_5, v_6, v_7, u_8)\big) = f\big(2,0,2, f(2,0,2,1)\big) = f(2,0,2,0) =$

$= 2 \Rightarrow v_8 = 2,$

$v_9 = T^1_{v_6,v_7,v_8} u_9 = f(v_6, v_7, v_8, u_9) = f(0,2,2,0) = 2 \Rightarrow v_9 = 2.$

Получаем следующий криптотекст: 122020222.

Заметим, что в условиях данного примера верен следующий факт:

$T^{-1}(x, y, z, -) = T^2(x, y, z, -).$

Воспользовавшись этим фактом, имеем следующую процедуру дешифрования:

$T^2_{l_1,l_2,l_3} v_1 = f(l_1, l_2, l_3, f(l_1, l_2, l_3, v_1)\,) = f\big(0,1,2, f(0,1,2,1)\big) = f(0,1,2,2) =$

$= {}^{(4,5)} f(0,1,2,1) = 0 \Rightarrow u_1 = 0,$

$T^1_{l_4,l_5,v_1} v_2 = f(l_4, l_5, v_1, v_2) = f(0,2,1,2) \; = 2 \Rightarrow u_2 = 2,$

$T^2_{l_6,v_1,v_2} v_3 = f\big(l_6, v_1, v_2, f(l_6, v_1, v_2, v_3)\big) = f\big(0,1,2, f(0,1,2,2)\big) = f(0,1,2,0) =$

$= {}^{(4,5)} f(0,1,2,2) = 1 \Rightarrow u_3 = 1,$

$T^1_{v_1,v_2,v_3} v_4 = f(v_1, v_2, v_3, v_4) = f(1,2,2,0) \; = 0 \Rightarrow u_4 = 0,$

$T^2_{v_2,v_3,v_4} v_5 = f\big(v_2, v_3, v_4, f(v_2, v_3, v_4, v_5)\big) = f(2,2,0, f(2,2,0,2)) = f(2,2,0,2) =$

$= {}^{(4,5)} f(2,2,0,2) = 2 \Rightarrow u_5 = 2,$

$T^1_{v_3,v_4,v_5} v_6 = f(v_3, v_4, v_5, v_6) = f(2,0,2,0) \; = 2 \Rightarrow u_6 = 2,$

$T^2_{v_4,v_5,v_6} v_7 = f\big(v_4, v_5, v_6, f(v_4, v_5, v_6, v_7)\big) = f(0,2,0, f(0,2,0,2)) = f(0,2,0,0) =$

$= {}^{(4,5)} f(0,2,0,2) = 1 \Rightarrow u_7 = 1,$

$T^1_{v_5,v_6,v_7} v_8 = f(v_5, v_6, v_7, v_8) = f(2,0,2,2) \; = 1 \Rightarrow u_8 = 1,$

$T^2_{v_6,v_7,v_8} v_9 = f\big(v_6, v_7, v_8, f(v_6, v_7, v_8, v_9)\big) = f\big(0,2,2, f(0,2,2,2)\big) = f(0,2,2,1) =$

$= {}^{(4,5)} f(0,2,2,2) = 0 \Rightarrow u_9 = 0.$

Получаем исходный открытый текст 021022110.

Замечание 2.5.6. Важной задачей в Обобщенном Алгоритме 2 является определение обратных трансляций в процедуре расшифровки.

2.6. Криптосистема Эль-Гамаля

Классическая система шифрования Тахера-Эль-Гамаля сформулирована на языке теории чисел с использованием умножения по модулю простого числа [159].

Схема Эль-Гамаля представляет собой криптосистему с открытым ключом, основанную на сложности вычисления дискретных логарифмов в конечном поле. Эта схема была предложена Эль-Гамалем в 1985 году. Он усовершенствовал систему Диффи-

Хеллмана и получил два алгоритма, которые использовались для шифрования и аутентификации. В отличие от RSA, алгоритм Эль-Гамаля не был запатентован и, следовательно, стал его более дешевой альтернативой [168].

Отправителем сообщений и их получателем могут быть отдельные лица, организации или технические системы. Это могут быть абоненты сети, пользователи компьютерной системы или абстрактные «стороны», участвующие в информационном взаимодействии. Но чаще участников отождествляют с людьми и заменяют формальными обозначениями A и B, как Алиса и Боб.

Предполагается, что сообщения передаются по так называемому «открытому» каналу связи, доступному для прослушивания некоторым третьим лицам. В этой среде легко реализуются как копирование, так и подмена передаваемых сообщений.

В криптографии обычно предполагается, что у лица, отправляющего сообщения или получающего их, есть некий противник E и этот противник может перехватывать сообщения, передаваемые по открытому каналу. Противником считается человек по имени Ева, имеющий в своем распоряжении мощное вычислительное оборудование и владеющий методами криптоанализа.

Перед отправкой сообщения по открытому каналу связи от A к B, A шифрует сообщение, а B, получив зашифрованное сообщение, расшифровывает его, восстанавливая исходный текст. Важно то, что Алиса и Боб могут договориться об используемом ими шифре не на открытом канале, а на особом «закрытом», недоступном для прослушивания противником канале. Такой «закрытый канал» можно организовать с помощью курьеров, либо Алиса и Боб могут обменяться шифрами при личной встрече. Обычно организация такого закрытого канала и передача по нему сообщений слишком затратны.

Перейдем теперь к анализу действий злоумышленника, пытающегося расшифровать сообщение и узнать секретный ключ. В криптографии принято, что противник может знать используемый алгоритм шифрования, характер передаваемых сообщений и перехваченный шифротекст, но не знает секретного ключа. Иногда это правило выглядит как «перестраховка», но такая «перестраховка» отнюдь не лишняя, если, скажем, передается распоряжение о переводе солидной суммы денег.

Допустим есть подписчики $A, B, C, \ldots$ которые хотят передавать друг другу зашифрованные сообщения, не имея никаких защищенных каналов связи. Код, предложенный Эль-Гамалем решает эту проблему, используя, в отличие от кода Шамира, только одну пересылку сообщений. Здесь используется схема Диффи-Хеллмана для

формирования общего секретного ключа для двух абонентов, передающих сообщение друг другу, а затем сообщение шифруется путем его умножения на этот ключ. Для каждого последующего сообщения секретный ключ пересчитывается.

Ключи генерируются первыми. Процедуру генерации условно можно разделить на пять этапов:

1) Сначала выбирается случайное простое число p.

2) Затем выбираем целое число g – первообразный корень для p.

Первообразный корень по модулю p это целое число g такое, что: $g^{\varphi(p)} \equiv 1(mod\ p)$ и $g^l \not\equiv 1(mod\ p)$ для $1 \leq l \leq \varphi(p)$, где $\varphi(p)$функция Эйлера (число натуральных чисел меньших p и взаимно простых с ним), значит первообразный корень является производным элементом мультипликативной группы кольца вычетов по модулю p.

Чтобы не проверять все числа l от 1 до $\varphi(p)$, можно проверить три условия:

a) является ли g взаимно простым с p, и если нет, то это не первообразный корень;

b) так как $\varphi(p)$ – всегда четное число, то для всех чисел $p > 2$, $\varphi(p)$ имеет хотя бы один простой делитель – простое число 2, поэтому для устранения значительного числа не корней достаточно проверить: $g^{\frac{\varphi(p)}{2}} \not\equiv 1\ (mod\ p)$ для числа, похожего на первообразный корень по модулю p. Если результат равен 1, то g не является корнем, если же результат равен $\varphi(p)$, тогда g возможно и является первообразным корнем;

c) далее следует убедиться, что $g^l \not\equiv 1\ (mod\ p)$, для всех $l = \frac{\varphi(p)}{s}$, где s – простой делитель числа $\varphi(p)$ полученный в результате его факторизации. Однако уже составлена таблица минимальных первообразных корней, которую можно использовать.

Числа p и g передаются абонентам в открытом виде.

3) Далее каждый абонент группы выбирает свой секретный ключ: c_i, $1 < c_i < p - 1$, где числа c_i и $(p - 1)$ взаимно просты.

4) Каждый подписчик вычисляет соответствующий открытый ключ d_i:

$$d_i \equiv g^{c_i}(mod\ p). \qquad (\mathbf{2.3})$$

5) Открытые ключи – это d_i, а закрытые ключи – это числа c_i.

Таблица 2.15. Пользовательские ключи в системе Эль-Гамаля

Абонент	Закрытый ключ	Открытый ключ
A	c_A	d_A
B	c_B	d_B
C	c_C	d_C
…	…	…

Покажем теперь, как пользователь A отправляет сообщение m абоненту B, которое представлено в виде числа $m < p$.

Шифрование. A формирует случайное число k (сеансовый ключ) , $1 \leq k \leq p-2$, причем k и $(p-1)$ взаимно просты. Далее вычисляются:

$$r \equiv g^k (mod\ p), \qquad (\mathbf{2.4})$$

$$e \equiv m \cdot d_B^k (mod\ p), \qquad (\mathbf{2.5})$$

и пара чисел (r, e) передается абоненту B.

Дешифрование. B, получив пару (r, e) и зная свой секретный ключ c_B вычисляет:

$$m' \equiv e \cdot r^{p-1-c_B}\ (mod\ p). \qquad (\mathbf{2.6})$$

Утверждение 2.6.1. (свойства шифра Эль-Гамаля).

1) Абонент B в результате получит сообщение $m' = m$;
2) Противник, зная p, g, d_B, r и e, не сможет вычислить m.

Доказательство. Подставим в (2.6) значение e из (2.5):

$$m' \equiv m \cdot d_B^k \cdot r^{p-1-c_B}\ (mod\ p).$$

Далее подставим (2.4) вместо r, и (2.3) −вместо d_B:

$$m' \equiv m \cdot (g^{c_B})^k \cdot (g^k)^{p-1-c_B} (mod\ p) \equiv\ m \cdot\ g^{c_B k + k(p-1) - k c_B}\ (mod\ p) \equiv$$

$$\equiv m \cdot\ g^{k(p-1)}\ (mod\ p).$$

По теореме Ферма: $g^{k(p-1)}\ (mod\ p) \equiv 1^k\ (mod\ p) = 1$. Первая часть утверждения доказана.

Для доказательства второй части заметим, что противник не может вычислить k в равенстве (2.4). Следовательно, он не может вычислить m в равенстве (2.5), так как m было умножено на неизвестное число. Противник также не может воспроизвести действия легитимного получателя сообщения, так как ему неизвестно секретное число c_B (вычисление c_B на основе (2.3) также является задачей дискретного логарифмирования).

Пример 2.6.2. Рассмотрим передачу сообщения $m = 218$ от A к B.

1) Возьмем случайное простое число $p = 293$.

2) Выберем целое число $g = 2$ как наименьший первообразный корень числа 293.

3) Затем подписчики выбирают свои секретные ключи. Пусть абонент A выбирает себе секретное число $c_A = 15$, а абонент B выбирает свое секретное число $c_B = 21$.

4) Каждый подписчик вычисляет соответствующий открытый ключ:

$d_A \equiv 2^{15}\ (mod\ 293) = 245, \qquad d_B \equiv 2^{21}\ (mod\ 293) = 151.$

Шифрование. Абонент A случайным образом выбирает число $k = 49$ и вычисляет числа r и e используя (2.4) и (2.5):

$r \equiv 2^{49}(mod\ 293) = 248,$

$e \equiv 218 \cdot 151^{49}(\ mod\ 293) \equiv 218 \cdot 165\ (mod\ 293\ ) = 224.$

После этого A отправляет зашифрованное сообщение в виде пары чисел $(248, 224)$.

Дешифрование. B получает пару (248,224) и вычисляет m' следующим образом:

$m' \equiv 224 \cdot 248^{293-1-21}\ (mod\ 293) \equiv 224 \cdot 248^{271}\ (mod\ 293) \equiv$

$\equiv 224 \cdot 103\ (mod\ 293) = 218.$

Так B смог расшифровать переданное сообщение 218.

Более того, любой абонент, знающий открытый ключ абонента B может отправлять ему сообщения, зашифрованные с помощью открытого ключа d_B. Однако, расшифровать эти сообщения может только абонент B с помощью известного только ему секретного ключа c_B. Объем шифра в 2 раза больше объема сообщения, но требуется только одна передача данных.

Шифр Шамира полностью решает проблему обмена сообщениями, когда абоненты могут использовать только открытые линии связи. Однако это сообщение трижды отправляется от одного абонента к другому, что является его недостатком. Шифр Эль-Гамаля позволяет решить ту же задачу за одну передачу данных, но объем передаваемого шифротекста в 2 раза превышает размер сообщения.

Поскольку в схему Эль-Гамаля вводится случайная величина k шифр Эль-Гамаля можно назвать шифром многозначной подстановки. Из-за случайности выбора числа k, такую схему также называют вероятностной схемой шифрования. Вероятностный характер шифрования является его преимуществом, поскольку такие схемы имеют лучшую стойкость ко взлому, чем схемы с конкретным процессом шифрования.

Недостатком схемы шифрования Эль-Гамаля является то, что длина зашифрованного текста удваивается по сравнению с исходным текстом. В этой схеме

необходимо использовать разные значения случайной величины k для шифрования разных сообщений. Для вероятностной схемы шифрования само сообщение m и ключ не определяют зашифрованный текст однозначно.

Схему Эль-Гамаля можно сформулировать в терминах кольца вычетов по модулю p или на языке поля Галуа $GF(p)$. Можно использовать понятие действия группы автоморфизмов циклической группы $(Z_p, +)$ на эту группу. Понятно, что разные точки зрения на одну и ту же математическую идею приводят к разным обобщениям.

Пусть $(Z_p, +)$ циклическая группа вычетов большого простого порядка относительно сложения вычетов, а $a -$ образующий элемент группы$(Z_{p-1}, \cdot)$, где:

$(Z_{p-1}, \cdot) \cong Aut(Z_p, +)(gcd(a, p-1) = 1)$.

Ключи Алисы будут следующие:

Открытый ключ: p, a и $a^m, m \in \mathbb{N}$.

Закрытый ключ: m.

Шифрование. Чтобы отправить сообщение $s \in (Z_{(p-1)}, \cdot)$ Боб вычисляет a^rи a^{mr} для случайного числа $r \in \mathbb{N}$ (иногда число r называют эфемерным ключом [3]).

Зашифрованный текст будет выглядеть так: $(a^r; a^{mr} \cdot s)$.

Дешифрование. Алиса зная m и получив зашифрованный текст $(a^r; a^{mr} \cdot s)$ вычисляет a^{mr} из a^r затем вычисляет a^{-mr}, и из $a^{mr} \cdot s$ находит искомое сообщение s.

Пример 2.6.3. Рассмотрим передачу сообщения $s = 155$ от B к A.

Алиса выбирает числа $p = 163,\ a = 5,\ m = 79$ и вычисляет:

$a^m \equiv 5^{79}(mod\ 163) \equiv 13\ (mod\ 163)$.

Ее открытый ключ: $(p, a^m) = (163,13)$ и закрытый ключ: $m = 79$. На практике этот выбор осуществляется в центре распределения ключей.

Боб хочет отправить сообщение s Алисе. Он выбирает случайное целое число $r = 47$ и шифрует сообщение $s = 155$ как $(a^m)^r \cdot s$. Боб получает:

$(5^{47}, 13^{47} \cdot 155)(mod\ 163) \equiv (17{,}141 \cdot 155)(mod\ 163) \equiv (17,13)$.

Он отправляет зашифрованное сообщение (17,13) Алисе. Алиса получает это сообщение и, используя свой закрытый ключ $m = 79$, расшифровывает его следующим образом:

$(17^{-79} \cdot 13)(mod\ 163) \equiv (17^{162-79} \cdot 13)(mod\ 163) \equiv (17^{83} \cdot 13)(mod\ 163) \equiv$

$\equiv (37 \cdot 13)(mod\ 163) = 155.$

Сложность системы шифрования Эль-Гамаля основана на сложности задачи дискретного логарифмирования. Эта система не защищена от атаки выбранным зашифрованным текстом [160]. Криптосистема Эль-Гамаля обычно используется в гибридной криптосистеме, где для шифрования самого сообщения и для шифрования ключа используется симметричная криптосистема.

2.7. Аналог схемы Эль-Гамаля, основанный на алгоритме Марковского

Приведем аналог системы шифрования Эль-Гамаля на основе алгоритма Марковского [45].

Пусть (Q, f) некоторая бинарная квазигруппа и $T = (\alpha, \beta, \gamma)$ ее изотопия.

Ключи Алисы:

Открытый ключ: $(Q, f), T, T^{(m,n,k)} = (\alpha^m, \beta^n, \gamma^k), m, n, k \in \mathbb{N}$ и алгоритм Марковского.

Закрытый ключ: m, n, k.

Ключи Боба:

Открытый ключ: $(Q, f), T, T^{(r,s,t)} = (\alpha^r, \beta^s, \gamma^t), r, s, t \in \mathbb{N}$ и алгоритм Марковского.

Закрытый ключ: r, s, t.

Шифрование.

Чтобы отправить сообщение $b \in (Q, f)$ Боб вычисляет $T^{(r,s,t)}$ для тройки случайных чисел $r, s, t \in \mathbb{N}$.

Затем, зная $T^{(m,n,k)}$ он вычисляет $T^{(mr,ns,kt)}$ и $(T^{(mr,ns,kt)}(Q, f))$.

Чтобы зашифровать сообщение b Боб использует известный Алисе алгоритм Марковского.

Получает зашифрованный текст и свою степенную изотопию: $(T^{(r,s,t)}, (T^{(mr,ns,kt)}(Q, f))b)$.

Дешифрование.

Алиса, зная числа m, n, k, и получив зашифрованный текст $(T^{(r,s,t)}, (T^{(mr,ns,kt)}(Q, f))b)$, вычисляет $(T^{(mr,ns,kt)}(Q, f))^{-1}$ используя $T^{(r,s,t)}$ и расшифровывает сообщение b.

Пример 2.7.1. Пусть $(Q, f) -$ бинарная квазигруппа, заданная следующей таблицей Кэли:

Таблица 2.16. Таблица Кэли квазигруппы$(Q, \cdot)$

·	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	5	2	6	4	0	3	1
1	1	6	5	3	4	2	0
2	0	5	4	6	3	1	2
3	4	1	3	0	2	6	5
4	2	4	0	1	6	5	3
5	6	3	1	2	5	0	4
6	3	0	2	5	1	4	6

и $T = (\alpha, \beta, \gamma)$ ее изотопия, где:

$\alpha = (2\ 3\ 4)\ (0\ 5\ 1\ 6) -$ соответствует перестановке строк квазигрупповой таблицы*;*

$\beta = (0\ 3\ 2\ 1)\ (5\ 6) -$ соответствует перестановке столбцов квазигрупповой таблицы, полученной после применения α*;*

$\gamma = (1\ 2\ 3\ 6\ 0\ 5\ 4) -$ подстановка, примененная к таблице, полученной после применения β. Для подстановки γ мы получаем обратную $\gamma^{-1} = (6\ 3\ 2\ 1\ 4\ 5\ 0)$.

Результат применения изотопии можно представить в виде следующей таблицы:

Таблица 2.17. Таблицы Кэли квазигрупп после действия подстановок $(\alpha, \beta, \gamma^{-1})$

α (·)	**0**	**1**	**2**	**3**	**4**	**5**	**6**		β (·)	**0**	**1**	**2**	**3**	**4**	**5**	**6**		γ^{-1} (·)	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	6	3	1	2	5	0	4		**0**	2	6	3	1	5	4	0		**0**	1	3	2	4	0	5	6
1	3	0	2	5	1	4	6		**1**	5	3	0	2	1	6	4		**1**	0	2	6	1	4	3	5
2	4	1	3	0	2	6	5		**2**	0	4	1	3	2	5	6		**2**	6	5	4	2	1	0	3
3	2	4	0	1	6	5	3		**3**	1	2	4	0	6	3	5		**3**	4	1	5	6	3	2	0
4	0	5	4	6	3	1	2		**4**	6	0	5	4	3	2	1		**4**	3	6	0	5	2	1	4
5	1	6	5	3	4	2	0		**5**	3	1	6	5	4	0	2		**5**	2	4	3	0	5	6	1
6	5	2	6	4	0	3	1		**6**	4	5	2	6	0	1	3		**6**	5	0	1	3	6	4	2

Ключи Алисы будут следующие:

Закрытый ключ: $m = 3, n = 6, k = 5.$

Открытый ключ: $(Q, f), T, T^{(3,6,5)} = (\alpha^3, \beta^6, \gamma^5)$, где:

$\alpha^3 = (0\ 6\ 1\ 5),\ \beta^6 = (0\ 2)\ (1\ 3),\ \gamma^5 = (0\ 3\ 1\ 5\ 6\ 2\ 4),\ (\gamma^5)^{-1} = (0\ 4\ 2\ 6\ 5\ 1\ 3),$

и алгоритм Марковского.

В результате получаем следующие таблицы Кэли:

Таблица 2.18. Таблицы Кэли квазигрупп после действия подстановок $((\alpha^3, \beta^6, (\gamma^5)^{-1})$

α^3 (·)	0	1	2	3	4	5	6
0	3	0	2	5	1	4	6
1	6	3	1	2	5	0	4
2	0	5	4	6	3	1	2
3	4	1	3	0	2	6	5
4	2	4	0	1	6	5	3
5	5	2	6	4	0	3	1
6	1	6	5	3	4	2	0

β^6 (·)	0	1	2	3	4	5	6
0	2	5	3	0	1	4	6
1	1	2	6	3	5	0	4
2	4	6	0	5	3	1	2
3	3	0	4	1	2	6	5
4	0	1	2	4	6	5	3
5	6	4	5	2	0	3	1
6	5	3	1	6	4	2	0

$(\gamma^5)^{-1}$ (·)	0	1	2	3	4	5	6
0	6	1	0	4	3	2	5
1	3	6	5	0	1	4	2
2	2	5	4	1	0	3	6
3	0	4	2	3	6	5	1
4	4	3	6	2	5	1	0
5	5	2	1	6	4	0	3
6	1	0	3	5	2	6	4

Шифрование. Для отправки сообщения $b = 630512403$ Боб зная основную изотопию

$$T = (\alpha, \beta, \gamma) : \alpha = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 5\,6\,3\,4\,2\,1\,0 \end{pmatrix};\ \beta = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 3\,0\,1\,2\,4\,6\,5 \end{pmatrix};\ \gamma = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 5\,2\,3\,6\,1\,4\,0 \end{pmatrix},$$

вычисляет изотопию $T^{(r,s,t)}$ для случайно выбранных чисел $r = 5, s = 3, t = 6$, т.е. $T^{(5,3,6)}$:

$$\alpha^5 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 5\,6\,4\,2\,3\,1\,0 \end{pmatrix};\ \beta^3 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 1\,2\,3\,0\,4\,6\,5 \end{pmatrix};\ \gamma^6 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 6\,4\,1\,2\,5\,0\,3 \end{pmatrix}.$$

В данном примере для $T^{(5,3,6)}$ получим:

$\alpha^5 = (0\,5\,1\,6)\,(2\,4\,3)$; $\beta^3 = (0\,1\,2\,3)\,(5\,6)$; $\gamma^6 = (0\,6\,3\,2\,1\,4\,5)$.

Затем он вычисляет $T^{(mr,ns,kt)}$ используя открытый ключ:

$T^{(m,n,k)} = (\alpha^m, \beta^n, \gamma^k) = (\alpha^*, \beta^*, \gamma^*)$:

$$\alpha^* = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 6\,5\,2\,3\,4\,0\,1 \end{pmatrix};\ \beta^* = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 2\,3\,0\,1\,4\,5\,6 \end{pmatrix};\ \gamma^* = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 3\,5\,4\,1\,0\,6\,2 \end{pmatrix}.$$

Боб возводит эти подстановки в степени $r = 5,\ s = 3,\ t = 6$ соответственно, и получает:

$$(\alpha^*)^5 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 6\,5\,2\,3\,4\,0\,1 \end{pmatrix};\ (\beta^*)^3 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 2\,3\,0\,1\,4\,5\,6 \end{pmatrix};\ (\gamma^*)^6 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 4\,3\,6\,0\,2\,1\,5 \end{pmatrix},$$ или

$\alpha^{5m} = (0\,6\,1\,5)$; $\beta^{3n} = (0\,2)\,(1\,3)$; $\gamma^{6k} = (0\,4\,2\,6\,5\,1\,3)$, $(\gamma^{6k})^{-1} = (0\,3\,1\,5\,6\,2\,4)$.

В результате применения новой изотопии $T^{(5m,3n,6k)}$ к квазигруппе (Q, f) получим:

Таблица 2.19. Таблицы Кэли квазигрупп после действия подстановок

$(\alpha^{5m}, \beta^{3n}, (\gamma^{6k})^{-1})$

α^{5m} (·)	0	1	2	3	4	5	6
0	3	0	2	5	1	4	6
1	6	3	1	2	5	0	4

β^{3n} (·)	0	1	2	3	4	5	6
0	2	5	3	0	1	4	6
1	1	2	6	3	5	0	4

$(\gamma^{6k})^{-1}$ (·)	0	1	2	3	4	5	6
0	4	6	1	3	5	0	2
1	5	4	2	1	6	3	0

2	0	5	4	6	3	1	2		**2**	4	6	0	5	3	1	2		**2**	0	2	3	6	1	5	4
3	4	1	3	0	2	6	5		**3**	3	0	4	1	2	6	5		**3**	1	3	0	5	4	2	6
4	2	4	0	1	6	5	3		**4**	0	1	2	4	6	5	3		**4**	3	5	4	0	2	6	1
5	5	2	6	4	0	3	1		**5**	6	4	5	2	0	3	1		**5**	2	0	6	4	3	1	5
6	1	6	5	3	4	2	0		**6**	5	3	1	6	4	2	0		**6**	6	1	5	2	0	4	3

Чтобы получить $(T^{(mr,ns,kt)}(Q,f)b$, Боб использует алгоритм Марковского, известный Алисе, с известным значением лидера $l = 3$, тогда зашифрованный текст для $b = 630512403$ будет выглядеть так:

$v_1 = 3 \circ 6 = 6, \quad v_2 = 6 \circ 3 = 2, \quad v_3 = 2 \circ 0 = 0,$

$v_4 = 0 \circ 5 = 0, \quad v_5 = 0 \circ 1 = 6, \quad v_6 = 6 \circ 2 = 5,$

$v_7 = 5 \circ 4 = 3, \quad v_8 = 3 \circ 0 = 1, \quad v_9 = 1 \circ 3 = 1.$

$b' = 620065311.$

Дешифрование. Алиса, зная $m = 3, n = 6, k = 5$, получив изотопию $T^{(r,s,t)}$ и зашифрованный текст $(T^{(mr,ns,kt)}(Q,f))b) = 620065311$, сначала вычисляет изотопию $T^{(mr,ns,kt)}$ используя $T^{(r,s,t)} = T^{(**,**,**)}$:

$$\alpha^{**} = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 5\,6\,4\,2\,3\,1\,0 \end{pmatrix}; \quad \beta^{**} = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 1\,2\,3\,0\,4\,6\,5 \end{pmatrix}; \quad \gamma^{**} = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 6\,4\,1\,2\,5\,0\,3 \end{pmatrix}.$$

$$(\alpha^{**})^3 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 6\,5\,2\,3\,4\,0\,1 \end{pmatrix}; \quad (\beta^{**})^6 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 2\,3\,0\,1\,4\,5\,6 \end{pmatrix}; \quad (\gamma^{**})^5 = \begin{pmatrix} 0\,1\,2\,3\,4\,5\,6 \\ 4\,3\,6\,0\,2\,1\,5 \end{pmatrix}$$

В результате она получает ту же Таблицу 2.19, которую Боб получил в процессе шифрования. Для таблицы $(\gamma^{6k})^{-1}$ Алиса строит (23)-парастроф, который используется в алгоритме Марковского для дешифрования:

Таблица 2.20. Таблица Кэли квазигруппы $(Q, \backslash)$

\	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	5	2	6	3	0	4	1
1	6	3	2	5	1	0	4
2	0	4	1	2	6	5	3
3	2	0	5	1	4	3	6
4	3	6	4	0	2	1	5
5	1	5	0	4	3	6	2
6	4	1	3	6	5	2	0

Используя Таблицу 2.20 Алиса вычисляет b:

$u_1 = 3\backslash 6 = 6, \quad u_2 = 6\backslash 2 = 3, \quad u_3 = 2\backslash 0 = 0,$

$u_4 = 0\backslash 0 = 5, \quad u_5 = 0\backslash 6 = 1, \quad u_6 = 6\backslash 5 = 2,$

$u_7 = 5\backslash 3 = 4, \quad u_8 = 3\backslash 1 = 0, \quad u_9 = 1\backslash 1 = 3.$

$b = 630512403.$

В этом алгоритме изострофия [161] может быть использована вместо изотопии, алгоритм из [29] вместо алгоритма Марковского и n-арные ($n > 2$) квазигруппы [39] вместо бинарных. Обобщение схемы распределения открытых ключей Диффи-Хеллмана дано в [169]. Обобщение основано на понятиях левых и правых степеней элементов некоторых неассоциативных группоидов. Для медиальных квазигрупп этот подход реализуется в [46]. Протокол разработки общего секретного ключа на основе луп Муфанг приведен в [46]. Этот протокол представляет собой обобщение результатов из [170]. В [162] дискретная логарифмическая задача с лупами Муфанг сводится к такой же задаче над конечными простыми полями. Другое обобщение схемы Эль-Гамаля, основанное на квазиавтоморфизмах квазигрупп, представлено в [46].

2.8. Выводы по Главе 2

Вторая глава посвящена разработке криптографических и алгебраических алгоритмов на основе алгоритма Марковского. Разработаны программы, демонстрирующие работу алгоритма Марковского для левой и правой бинарных квазигрупп. Обобщения алгоритма Марковского построены на основе n -арных группоидов: Обобщенный Алгоритм 1 и Обобщенный Алгоритм 2 (с использованием трансляций). Разработаны программы, реализующие работу этих алгоритмов, описаны специфика и особенности этих программ.

На основании исследования, проведенного в Главе 2 и полученных результатов можно сделать следующие выводы.:

1) Разработаны обобщенные алгоритмы Марковского для левой и правой квазигрупп и программы для их реализации (Приложение 2), которые имеют свои особенности и преимущества [171, 172];
2) Построен обобщенный алгоритм Марковского для n -арного группоида, обратимого на одном фиксированном месте – Обобщенный Алгоритм 1, и разработаны программы, реализующие работу этого алгоритма [171, 173, 174];
3) Обобщенный алгоритм Марковского для n-арного группоида, обратимого на одном фиксированном месте, построен с использованием трансляций любых

степеней – Обобщенный Алгоритм 2 и были разработаны программы, реализующие работу этого алгоритма [174-176];

4) Сравнивая построенные два алгоритма, видим, что общее количество необходимых лидеров для первого алгоритма будет: $(n-1)^2$, а для второго алгоритма искомое количество лидеров будет равно: $\frac{(n-1)n}{2}$. Второе число меньше первого на величину: $(n-1)\left(\frac{n}{2}-1\right)$, что говорит о преимуществе второго алгоритма перед первым (особенно при росте числа n) [174];
5) Обобщенный Алгоритм 2 будет намного сложнее, если помимо первой и второй степеней трансляций использовать другие более высокие степени. Особый интерес представляет определение обратных трансляций для тех, которые используются в Обобщенном Алгоритме 2 [174];
6) Рассмотрен аналог системы шифрования Эль-Гамаля на основе алгоритма Марковского и изучены его особенности. Этот алгоритм находится в стадии доработки и планируются другие его модификации [177-180].

На основании изложенного была сформулирована цель дальнейшего исследования, заключающаяся в проведении криптоанализа всех алгоритмов, построенных в данной главе.

Для достижения поставленной цели были сформулированы следующие задачи:

- проведение атак на шифры, построенные во второй главе;
- сравнительный анализ проведенных атак;
- подбор оптимальных вариантов текста для всех изученных типов атак.

В этой главе достигаются цели, связанные с применением алгоритма Марковского к построению криптографических и алгебраических алгоритмов. Результаты, представленные в Главе 2, были опубликованы в [171-180].

3. КРИПТОАНАЛИЗ НЕКОТОРЫХ ПОТОКОВЫХ ШИФРОВ (БИНАРНЫЙ СЛУЧАЙ)

Потребность в случайных и псевдослучайных последовательностях возникает во многих приложениях, например, в моделировании и в криптографии. Псевдослучайные последовательности являются основой поточных шифров. Целью проектирования поточных шифров является эффективное создание псевдослучайных последовательностей– ключевых потоков [181, 182]. Генераторы псевдослучайных чисел, подходящие для использования в криптографических приложениях, могут нуждаться в более строгих требованиях, чем для других приложений.

Использование квазигрупп в криптографии не очень распространено. Несмотря на это, в последние годы появились различные криптосистемы на основе квазигрупп [183], где показано, что использование квазигрупп при проектировании S-блоков может открыть новые пути в проектировании блочных шифров.

С. Марковски, Э. Оходкова, В. Снасель, Д. Глигороски и С. Андова предложили новый поточный шифр для шифрования файловой системы [167, 25]. Шифр имеет очень большое пространство ключей и считается устойчивым к любой атаке.

М. Войвода провел криптоанализ системы кодирования файлов на основе бинарных квазигрупп [184, 34] и показал, как взломать этот шифр, используя атаку только зашифрованным текстом [185]. В его статье описан исследуемый поточный шифр, а также обобщены результаты, изученные на тот момент по шифрам в области криптоанализа.

В этой главе будут проведены следующие исследования:

- показано, как работает криптоанализ в случаях применения левой и правой квазигрупп;
- рассмотрены модификации криптографических атак, построенных М. Войводой для квазигрупп;
- проведен сравнительный анализ и выявлены положительные и отрицательные моменты в исследуемых атаках;
- описаны некоторые атаки на шифр с использованием выбранного шифротекста и открытого текста, построенного на основе обобщенных алгоритмов Марковского.

3.1. Атаки выбранным шифротекстом на шифр Марковского, основанный на квазигруппе.

Пусть $(Q,*)$ – конечная квазигруппа. Символы открытого текста $u_1, u_2, \dots, u_k$ и зашифрованного текста $v_1, v_2, \dots, v_k$ представлены элементами из Q, т.е. $u_i \in Q,\ 1 \le i \le k$. Ключом этого шифра является операция " $*$ " определенная на множестве Q.

Существует как минимум $(n!\,(n-1)!\,(n-2)!\dots 2!)$ латинских квадратов порядка n. Если предположить, что $Q = \{0,1,\dots,255\}$ (т. е. данные представлены 8 битами = 1 байт), то имеется по крайней мере 1058000 квазигрупп порядка 256. Пространство ключей чрезвычайно велико. Заявлялось, что такой шифр устойчив к любой атаке [25] , хотя авторы изучали только стойкость к атаке методом грубой силы и провели некоторые статистические тесты этого шифра. С точки зрения криптоанализа хороший шифр должен быть устойчив хотя бы к известным типам атак [185].

Предположим, что известен порядок квазигруппы $(Q,*)$, где $Q = \{q_1, q_2, \dots, q_n\}$. Для каждого $i,\ 1 \le i \le n$ криптоаналитик определяет количество вхождений пар элементов $q_i q_j, 1 \le i \le n, 1 \le j \le n$. Если зашифрованный текст достаточно велик, для каждого q_i, $1 \le i \le n$, полученное количество вхождений пар элементов может быть сопоставлено с известными частотами вхождений отдельных символов из используемого языка. Тогда, криптоаналитик может построить таблицу Кэли квазигруппы $(Q,\ \backslash)$ и расшифровать сообщение. Восстановление ключа, т. е. таблицы Кэли, квазигруппы $(Q,\ *)$ по квазигруппе $(Q,\ \backslash)$ не представляет сложности.

Однако сопоставление полученного числа вхождений пар элементов с известными частотами вхождений отдельных символов используемого языка может привести к некоторым ошибкам в восстановлении квазигруппы $(Q,\ \backslash)$, либо из-за малой длины анализируемого зашифрованного текста, либо из-за специфических свойств используемого языка. Наилучший подход состоит в том, чтобы сопоставить только очевидные пары элементов, а затем частично расшифровать зашифрованный текст. Из частично расшифрованного сообщения возможно найти какие-то другие ячейки в таблице Кэли квазигруппы $(Q,\ \backslash)$. Это приводит к повторной расшифровке и построению таблицы Кэли квазигруппы $(Q,\ \backslash)$. Можно также использовать известные результаты из критических множеств [186].

Предположим, что криптоаналитик имеет доступ к устройству дешифрования, загруженному ключом. Затем он может построить следующий зашифрованный текст:

$$q_1q_1q_1q_2q_1q_3 \cdots q_1q_n$$
$$q_2q_1q_2q_2q_2q_3 \cdots q_2q_n \cdots$$
$$q_nq_1q_nq_2q_nq_3 \cdots q_nq_n.$$

и ввести его в устройство дешифрования.

Устройство дешифрования дает следующий открытый текст:

$$l\backslash q_1q_1\backslash q_1q_1\backslash q_1q_1\backslash q_1q_2\backslash q_1q_1\backslash q_3 \cdots q_1\backslash q_n$$
$$q_n\backslash q_2q_2\backslash q_1q_1\backslash q_2q_2\backslash q_2q_2\backslash q_2q_2\backslash q_3 \cdots q_2\backslash q_n \cdots$$
$$q_n\backslash q_nq_n\backslash q_1q_1\backslash q_nq_n\backslash q_2q_2\backslash q_nq_n\backslash q_3 \cdots q_n\backslash q_n.$$

В результате таблица Кэли операции "\ " будет полностью найдена. Шифротекст используемый М. Войводой в этой атаке состоит из $2n^2$ символов, но можно построить более короткий зашифрованный текст. Главное требование М. Войводы состоит в том, чтобы в шифротексте фигурировали все пары соседних элементов.

Для полной реконструкции таблицы Кэли квазигруппы $(Q, \backslash)$ достаточно ввести только $2n^2 - 4n + 1$ символов вместо $2n^2$. Эти атаки будем называть *усеченными атаками Войводы*.

Замечание 3.1.1. Сравнивая количество символов, используемых в атаке Войводы и усеченной атаке Войводы, получаем следующее предельное соотношение:

$$\lim_{n\to\infty} \frac{2n^2}{2n^2 - 4n + 1} = 1.$$

Рассмотрим новый тип атаки, которую будем называть *модифицированной атакой*. В процедуре расшифровки используется следующий текст:

$$q_1q_1q_2q_2q_3q_3 \cdots q_{n-2}q_{n-2}q_{n-1}q_{n-1}q_nq_n$$
$$q_2q_1q_3q_2q_4q_3 \cdots q_{n-1}q_{n-2}q_nq_{n-1}q_1q_n$$
$$q_3q_1q_4q_2q_5q_3 \cdots q_nq_{n-2}q_1q_{n-1}q_2q_n \cdots .$$

Устройство дешифрования выдает на выходе следующий открытый текст:

$$l\backslash q_1 \;\; q_1\backslash q_1 \;\; q_1\backslash q_2 \;\; q_2\backslash q_2 \;\; \cdots q_n\backslash q_n$$
$$q_n\backslash q_2 \;\; q_2\backslash q_1 \;\; q_1\backslash q_3 \;\; q_3\backslash q_2 \;\; \cdots q_1\backslash q_n$$
$$q_n\backslash q_3 \;\; q_3\backslash q_1 \;\; q_1\backslash q_4 \;\; q_4\backslash q_2 \;\; \cdots q_2\backslash q_n \cdots .$$

Последний символ зависит от четности порядка квазигруппы, а именно, если $n -$ нечетное число, то последней операцией будет: $q_k\backslash q_n$, где $k = \left[\frac{n}{2}\right] + 1$. Если же $n -$ четное число, то последней операцией будет: $q_{\frac{n}{2}}\backslash q_n$.

Таблица Кэли операции "\" находится полностью, после чего легко найти таблицу Кэли операции " $*$ ". Представленная атака требует: $n^2 - 2(n-1)$ операций "\". При сравнении с атакой М. Войводы видим, что количество используемых символов сокращается на $(n+1)^2 - 3$ символов, что весьма существенно, особенно при увеличении числа n. И главное, это число не зависит от используемого лидера.

Замечание 3.1.2. Сравнивая количество символов, используемых в атаке Войводы, усеченной атаке Войводы и модифицированной атаке, имеем следующие предельные соотношения:

$$\lim_{n\to\infty} \frac{2n^2}{n^2-2n+2} = 2 \text{ и } \lim_{n\to\infty} \frac{2n^2-4n+1}{n^2-2n+2} = 2.$$

Если сравнить этот результат с результатом, полученным в Замечании 3.1.1, то видно явное преимущество модифицированной атаки.

Пример 3.1.3. Пусть $Q = \{q_1 = 0,\ q_2 = 1,\ q_3 = 2,\ q_4 = 3,\ q_5 = 4\}$ и пусть квазигруппа $(Q,\backslash)$ с помощью которой выполняется дешифровка, имеет следующую таблицу Кэли:

Таблица 3.1. Таблица Кэли квазигруппы $(\boldsymbol{Q},\ \backslash)$

\	**0**	**1**	**2**	**3**	**4**
0	1	2	0	4	3
1	0	1	2	3	4
2	4	3	1	0	2
3	3	0	4	2	1
4	2	4	3	1	0

Пусть $l = 3,\ l \in Q$.

Вводим следующий текст в устройство дешифрования:

$$\begin{array}{lll} \boldsymbol{q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5} & & \boldsymbol{0001020304} \\ \boldsymbol{q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5} & & \boldsymbol{1011121314} \\ \boldsymbol{q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5} & \text{или} & \boldsymbol{2021222324} \\ \boldsymbol{q_4}q_1q_4q_2q_4q_3q_4q_4q_4q_5 & & \boldsymbol{3}031323334 \\ q_5q_1q_5q_2q_5q_3q_5q_4q_5q_5 & & 4041424344. \end{array}$$

На выходе получаем:

Таблица 3.2. Процесс дешифрования

$u_1 = l\backslash q_1 =$ $= 3\backslash 0 = 3$	$u_{11} = q_5\backslash q_2 =$ $= 4\backslash 1 = 4$	$u_{21} = q_5\backslash q_3 =$ $= 4\backslash 2 = 3$	$\boldsymbol{u_{31} = q_5\backslash q_4 =}$ $\boldsymbol{= 4\backslash 3 = 1}$	$u_{41} = q_5\backslash q_5 =$ $= 4\backslash 4 = 0$
$u_2 = q_1\backslash q_1 =$ $= 0\backslash 0 = 1$	$u_{12} = q_2\backslash q_1 =$ $= 1\backslash 0 = 0$	$u_{22} = q_3\backslash q_1 =$ $= 2\backslash 0 = 4$	$u_{32} = q_4\backslash q_1 =$ $= 3\backslash 0 = 3$	$u_{42} = q_5\backslash q_1 =$ $= 4\backslash 0 = 2$
$u_3 = q_1\backslash q_1 =$ $= 0\backslash 0 = 1$	$u_{13} = q_1\backslash q_2 =$ $= 0\backslash 1 = 2$	$u_{23} = q_1\backslash q_3 =$ $= 0\backslash 2 = 0$	$u_{33} = q_1\backslash q_4 =$ $= 0\backslash 3 = 4$	$u_{43} = q_1\backslash q_5 =$ $= 0\backslash 4 = 3$
$u_4 = q_1\backslash q_2 =$ $= 0\backslash 1 = 2$	$u_{14} = q_2\backslash q_2 =$ $= 1\backslash 1 = 1$	$u_{24} = q_3\backslash q_2 =$ $= 2\backslash 1 = 3$	$u_{34} = q_4\backslash q_2 =$ $= 3\backslash 1 = 0$	$u_{44} = q_5\backslash q_2 =$ $= 4\backslash 1 = 4$
$u_5 = q_2\backslash q_1 =$ $= 1\backslash 0 = 0$	$u_{15} = q_2\backslash q_2 =$ $= 1\backslash 1 = 1$	$u_{25} = q_2\backslash q_3 =$ $= 1\backslash 2 = 2$	$u_{35} = q_2\backslash q_4 =$ $= 1\backslash 3 = 3$	$u_{45} = q_2\backslash q_5 =$ $= 1\backslash 4 = 4$
$u_6 = q_1\backslash q_3 =$ $= 0\backslash 2 = 0$	$u_{16} = q_2\backslash q_3 =$ $= 1\backslash 2 = 2$	$u_{26} = q_3\backslash q_3 =$ $= 2\backslash 2 = 1$	$u_{36} = q_4\backslash q_3 =$ $= 3\backslash 2 = 4$	$u_{46} = q_5\backslash q_3 =$ $= 4\backslash 2 = 3$
$u_7 = q_3\backslash q_1 =$ $= 2\backslash 0 = 4$	$u_{17} = q_3\backslash q_2 =$ $= 2\backslash 1 = 3$	$u_{27} = q_3\backslash q_3 =$ $= 2\backslash 2 = 1$	$u_{37} = q_3\backslash q_4 =$ $= 2\backslash 3 = 0$	$u_{47} = q_3\backslash q_5 =$ $= 2\backslash 4 = 2$
$u_8 = q_1\backslash q_4 =$ $= 0\backslash 3 = 4$	$u_{18} = q_2\backslash q_4 =$ $= 1\backslash 3 = 3$	$u_{28} = q_3\backslash q_4 =$ $= 2\backslash 3 = 0$	$u_{38} = q_4\backslash q_4 =$ $= 3\backslash 3 = 2$	$u_{48} = q_5\backslash q_4 =$ $= 4\backslash 3 = 1$
$u_9 = q_4\backslash q_1 =$ $= 3\backslash 0 = 3$	$u_{19} = q_4\backslash q_2 =$ $= 3\backslash 1 = 0$	$u_{29} = q_4\backslash q_3 =$ $= 3\backslash 2 = 4$	$u_{39} = q_4\backslash q_4 =$ $= 3\backslash 3 = 2$	$u_{49} = q_4\backslash q_5 =$ $= 3\backslash 4 = 1$
$u_{10} = q_1\backslash q_5 =$ $= 0\backslash 4 = 3$	$u_{20} = q_2\backslash q_5 =$ $= 1\backslash 4 = 4$	$u_{30} = q_3\backslash q_5 =$ $= 2\backslash 4 = 2$	$u_{40} = q_4\backslash q_5 =$ $= 3\backslash 4 = 1$	$u_{50} = q_5\backslash q_5 =$ $= 4\backslash 4 = 0.$

Разбив текст на пять блоков получим:

3112004433

4021123304

3403211042

1340340221

0234432110.

Строки таблицы Кэли квазигруппы $(Q, \backslash)$ отображаются последовательно на четных позициях. Однако для полной реконструкции таблицы Кэли для квазигруппы $(Q, \backslash)$ достаточно ввести всего 31 символ (они выделены жирным шрифтом) вместо 50.

Лидер l является решением уравнения: $l\backslash 0 = 3 \Rightarrow l = 3$. Зная таблицу для квазигруппы $(Q, \backslash)$, таблица квазигруппы, используемой для шифрования легко восстанавливается:

Таблица 3.3. Таблица Кэли квазигруппы $(Q,\ *)$

*	**0**	**1**	**2**	**3**	**4**
0	2	0	1	4	3
1	0	1	2	3	4
2	3	2	4	1	0
3	1	4	3	0	2
4	4	3	0	2	1

Теперь рассмотрим модифицированную атаку для данного примера и в устройство дешифрования введем следующий текст:

$q_1q_1q_2q_2q_3q_3q_4q_4q_5q_5$ или 0011223344

$q_2q_1q_3q_2q_4q_3q_5$ $1021324.$

Таблица 3.4. Процесс дешифрования

$u_1 = l\backslash q_1 = 3\backslash 0 = 3$	$u_7 = q_3\backslash q_4 = 2\backslash 3 = 0$	$u_{13} = q_1\backslash q_3 = 0\backslash 2 = 0$
$u_2 = q_1\backslash q_1 = 0\backslash 0 = 1$	$u_8 = q_4\backslash q_4 = 3\backslash 3 = 2$	$u_{14} = q_3\backslash q_2 = 2\backslash 1 = 3$
$u_3 = q_1\backslash q_2 = 0\backslash 1 = 2$	$u_9 = q_4\backslash q_5 = 3\backslash 4 = 1$	$u_{15} = q_2\backslash q_4 = 1\backslash 3 = 3$
$u_4 = q_2\backslash q_2 = 1\backslash 1 = 1$	$u_{10} = q_5\backslash q_5 = 4\backslash 4 = 0$	$u_{16} = q_4\backslash q_3 = 3\backslash 2 = 4$
$u_5 = q_2\backslash q_3 = 1\backslash 2 = 2$	$u_{11} = q_5\backslash q_2 = 4\backslash 1 = 4$	$u_{17} = q_3\backslash q_5 = 2\backslash 4 = 2$
$u_6 = q_3\backslash q_3 = 2\backslash 2 = 1$	$u_{12} = q_2\backslash q_1 = 1\backslash 0 = 0$	

Таким образом, вместо 50 символов будет использоваться только 17 символов:

$$31212102104003342.$$

Рассмотрим работу усеченных и модифицированных атак для квазигрупп порядка 6.

Пример 3.1.4. Пусть $Q = \{q_1 = 0,\ q_2 = 1,\ q_3 = 2,\ q_4 = 3,\ q_5 = 4,\ q_6 = 5\}$ и квазигруппа $(Q,\ \backslash)$, с помощью которой производится дешифрование, имеет следующую таблицу Кэли:

Таблица 3.5. Таблица Кэли квазигруппы $(Q,\ \backslash)$

\	**0**	**1**	**2**	**3**	**4**	**5**
0	2	0	3	1	4	5
1	5	3	1	0	2	4
2	1	4	0	3	5	2
3	0	2	4	5	1	3
4	3	5	2	4	0	1
5	4	1	5	2	3	0

Пусть $l = 2,\ l \in Q$.

Введем следующий текст в устройство дешифрования:

$q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5q_1q_6$ 000102030405
$q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5q_2q_6$ 101112131415
$q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5q_3q_6$ или 202122232425
$q_4q_1q_4q_2q_4q_3q_4q_4q_4q_5q_4q_6$ 303132333435
q_5 4.

Процесс дешифровки можно найти в Приложении 3, Таблица А3.4. На выходе получаем 49 символов. Разбив текст на блоки получим:

122053110435
150331402254
513410034522
201204355143
3.

В этом примере будут использоваться только первые 49 символов вместо 72. Лидер l является решением уравнения: $l \setminus 0 = 1 \Rightarrow l = 2$. Зная таблицу квазигруппы $(Q, \setminus)$, легко восстанавливается таблица используемая для шифрования:

Таблица 3.6. Таблица Кэли квазигруппы $(Q, *)$

*	**0**	**1**	**2**	**3**	**4**	**5**
0	1	3	0	2	4	5
1	3	2	4	1	5	0
2	2	0	5	3	1	4
3	0	4	1	5	2	3
4	4	5	2	0	3	1
5	5	1	3	4	0	2

Теперь воспользуемся модифицированной атакой. Если в устройство дешифрования вводится следующий текст:

$q_1q_1q_2q_2q_3q_3q_4q_4q_5q_5q_6q_6$ 001122334455
$q_2q_1q_3q_2q_4q_3q_5q_4q_6q_5q_1q_6$ или 102132435405
q_3q_1 20.

Процесс дешифрования для этой атаки можно просмотреть в Приложении 3, Таблица А3.5. Вместо 49 символов для успешной атаки использовано всего 26 символов:

120310351010

153404543335

51.

Криптографическая атака на поточный шифр использует предположение, что криптоаналитик знает статистику языка, на котором написано открытое текстовое сообщение.

3.2. Атаки выбранным открытым текстом на шифр Марковского, основанный на квазигруппе

Атака выбранным зашифрованным текстом и атака выбранным открытым текстом очень похожи [184], но последняя имеет свои особенности, на которые стоит обратить внимание. Предположим, что каждый символ в текстовом сообщении представлен одним элементом из квазигруппы, а порядок квазигруппы $(Q,*)$, где $Q = \{q_1, q_2, \dots, q_n\}$ равен n.

Считаем, что у криптоаналитика есть доступ к шифровальному устройству. В работе М. Войводы [185] для шифрования строится следующий текст:

$$q_1q_1; q_1q_2; q_1q_3; \dots q_1q_n;$$
$$q_2q_1; q_2q_2; q_2q_3; \dots q_2q_n; \dots$$
$$q_nq_1; q_nq_2; q_nq_3; \dots q_nq_n.$$

Этот текст вводится в шифровальное устройство дискретно по 2 символа. Благодаря этому вводу имеем следующий зашифрованный текст:

$$l * q_1 \;\; (l * q_1) * q_1; l * q_1 \;\; (l * q_1) * q_2; \dots l * q_1 \;\; (l * q_1) * q_n;$$
$$l * q_2 \;\; (l * q_2) * q_1; l * q_2 \;\; (l * q_2) * q_2; \dots l * q_2 \;\; (l * q_2) * q_n; \dots$$
$$l * q_n \;\; (l * q_n) * q_1; l * q_n \;\; (l * q_n) * q_2; \dots l * q_n \;\; (l * q_n) * q_n.$$

Таблица Кэли операции " $*$ ", определенной на Q, находится полностью. Открытый текст, используемый в атаке, состоит из $\boldsymbol{2n^2}$ символов, разделенных на пары. На выходе: на нечетной позиции выводится номер строки, а на четной позиции – элемент квазигруппы $(Q, *)$. В отличие от атаки выбранным шифротекстом, в этой атаке вывод строк не упорядочен.

Можно построить более короткий зашифрованный текст, состоящий из $\boldsymbol{2(n-1)^2}$ символов. Это будет *усеченная атака Войводы*. Преимущество этих атак в том, что они не зависят от используемого лидера.

Теперь рассмотрим вариант, когда символы запускаются в шифровальное устройство потоком, а именно как в случае атаки выбранным шифротекстом:

$$q_1\ q_1\ q_1\ q_2\ q_1\ q_3 \dots q_1\ q_n$$
$$q_2\ q_1\ q_2\ q_2\ q_2\ q_3 \dots q_2\ q_n \dots$$
$$q_n q_1\ q_n q_2\ q_n q_3 \dots q_n q_n \dots .$$

Шифровальное устройство выдает на выходе следующий зашифрованный текст:

$v_1 = l * q_1, v_2 = v_1 * q_1, v_3 = v_2 * q_1, v_4 = v_3 * q_2, v_5 = v_4 * q_1, v_6 = v_5 * q_3, \dots$

$v_{2n} = v_{2n-1} * q_n, v_{2n+1} = v_{2n} * q_2 , \dots$

$v_{4n} = v_{4n-1} * q_n , \dots$

$v_{2n^2-2n} = v_{2n^2-2n-1} * q_n , \dots$

$v_{2n^2} = v_{2n^2-1} * q_n \dots .$

Будем называть эту атаку *потоковой атакой Войводы*. В ней количество используемых символов меньше, чем в предыдущих двух атаках, но результат зависит от используемого лидера. Это недостаток потоковой атаки.

Для каждого случая можно подобрать *потоковую атаку с минимальным количеством символов*, но задача эта достаточно сложная. Для квазигруппы порядка n необходимое минимальное количество символов равно $\boldsymbol{n(n-2)+2}$.

Теперь рассмотрим *модифицированную атаку с использованием выбранного открытого текста с дискретным вводом символов*:

$$q_1q_1; q_2q_2; q_3q_3; \dots q_{n-2}q_{n-2}; q_{n-1}q_{n-1}; q_nq_n;$$
$$q_2q_1; q_3q_2; q_4q_3; \dots q_{n-1}q_{n-2}; q_nq_{n-1}\ ; q_1q_n;$$
$$q_3q_1; q_4q_2; q_5q_3; \dots q_nq_{n-2}; q_1q_{n-1}; q_2q_n \dots .$$

Открытый текст, используемый в атаке, состоит из $\boldsymbol{2\ (n-1)^2}$ символов, разделенных на пары. Шифровальное устройство выдает на выходе следующий зашифрованный текст:

$v_1 = l * q_1 ,$ $v_2 = v_1 * q_1 ;$

$v_3 = l * q_2,$ $v_4 = v_3 * q_2;$

$v_5 = l * q_3,$ $v_6 = v_5 * q_3; \dots,$

$v_{2n-1} = l * q_n,$ $v_{2n} = v_{2n-1} * q_n ; \dots$

$v_{2(n-1)^2-1} = l * q_{n-1} ,$ $v_{2(n-1)^2} = v_{2(n-1)^2-1} * q_1.$

Наиболее удобными являются атаки с помощью усеченного текста Войводы и модифицированная атака, в которой символы вводятся дискретно по 2 и которые не зависят от выбранного лидера.

Пример 3.2.1. Пусть $Q = \{q_1 = 0,\ q_2 = 1,\ q_3 = 2,\ q_4 = 3,\ q_5 = 4\}$ и пусть дана квазигруппа $(Q,*)$ (Таблица 3.3, Пример 3.1.3), с помощью которой производится шифрование. Пусть лидер $l = 3,\ l \in Q$.

1) Рассмотрим атаку Войводы выбранным открытым текстом. Введем следующий текст в устройство шифрования:

$$\begin{array}{ll} \boldsymbol{q_1q_1\ q_1q_2\ q_1q_3\ q_1q_4}\ q_1q_5 & \mathbf{00\ 01\ 02\ 03}\ 04 \\ \boldsymbol{q_2q_1\ q_2q_2\ q_2q_3\ q_2q_4}\ q_2q_5 & \mathbf{10\ 11\ 12\ 13}\ 14 \\ \boldsymbol{q_3q_1\ q_3q_2\ q_3q_3\ q_3q_4}\ q_3q_5 \quad \text{или} & \mathbf{20\ 21\ 22\ 23}\ 24 \\ \boldsymbol{q_4q_1\ q_4q_2\ q_4q_3\ q_4q_4}\ q_4q_5 & \mathbf{30\ 31\ 32\ 33}\ 34 \\ q_5q_1\ q_5q_2\ q_5q_3\ q_5q_4\ q_5q_5 & 40\ 41\ 42\ 43\ 44. \end{array}$$

Текст вводится в шифровальное устройство дискретно по 2 символа. На выходе имеем следующий зашифрованный текст:

Таблица 3.7. Процесс шифрования

$v_1 = l * q_1 = 3 * 0 = 1,$ $v_2 = (l * q_1) * q_1 = 1 * 0 = 0;$	**10**	$v_{31} = l * q_4 = 3 * 3 = 0,$ $v_{32} = (l * q_4) * q_1 = 0 * 0 = 2;$	02
$v_3 = l * q_1 = 3 * 0 = 1,$ $v_4 = (l * q_1) * q_2 = 1 * 1 = 1;$	**11**	$v_{33} = l * q_4 = 3 * 3 = 0,$ $v_{34} = (l * q_4) * q_2 = 0 * 1 = 0;$	00
$v_5 = l * q_1 = 3 * 0 = 1,$ $v_6 = (l * q_1) * q_3 = 1 * 2 = 2;$	**12**	$v_{35} = l * q_4 = 3 * 3 = 0,$ $v_{36} = (l * q_4) * q_3 = 0 * 2 = 1;$	01
$v_7 = l * q_1 = 3 * 0 = 1,$ $v_8 = (l * q_1) * q_4 = 1 * 3 = 3;$	**13**	$v_{37} = l * q_4 = 3 * 3 = 0,$ $v_{38} = (l * q_4) * q_4 = 0 * 3 = 4;$	04
$v_9 = l * q_1 = 3 * 0 = 1,$ $v_{10} = (l * q_1) * q_5 = 1 * 4 = 4;$	14 (лишний)	$v_{39} = l * q_4 = 3 * 3 = 0,$ $v_{40} = (l * q_4) * q_5 = 0 * 4 = 3;$	03 (лишний)
$v_{11} = l * q_2 = 3 * 1 = 4,$ $v_{12} = (l * q_2) * q_1 = 4 * 4 = 4;$	**44**	$v_{41} = l * q_5 = 3 * 4 = 2,$ $v_{42} = (l * q_5) * q_1 = 2 * 0 = 3;$	23 (лишний)
$v_{13} = l * q_2 = 3 * 1 = 4,$ $v_{14} = (l * q_2) * q_2 = 4 * 1 = 3;$	**43**	$v_{43} = l * q_5 = 3 * 4 = 2,$ $v_{44} = (l * q_5) * q_2 = 2 * 1 = 2;$	22 (лишний)
$v_{15} = l * q_2 = 3 * 1 = 4,$ $v_{16} = (l * q_2) * q_3 = 4 * 2 = 0;$	**40**	$v_{45} = l * q_5 = 3 * 4 = 2,$ $v_{46} = (l * q_5) * q_3 = 2 * 2 = 4;$	24 (лишний)
$v_{17} = l * q_2 = 3 * 1 = 4,$ $v_{18} = (l * q_2) * q_4 = 4 * 3 = 2;$	**42**	$v_{47} = l * q_5 = 3 * 4 = 2,$ $v_{48} = (l * q_5) * q_4 = 2 * 3 = 1;$	21 (лишний)
$v_{19} = l * q_2 = 3 * 1 = 4,$ $v_{20} = (l * q_2) * q_5 = 4 * 4 = 1;$	41 (лишний)	$v_{49} = l * q_5 = 3 * 4 = 2,$ $v_{50} = (l * q_5) * q_5 = 2 * 4 = 0;$	20 (лишний)

$v_{21} = l * q_3 = 3 * 2 = 3,$ $v_{22} = (l * q_3) * q_1 = 3 * 0 = 1;$	**31**		
$v_{23} = l * q_3 = 3 * 2 = 3,$ $v_{24} = (l * q_3) * q_2 = 3 * 1 = 4;$	**34**		
$v_{25} = l * q_3 = 3 * 2 = 3,$ $v_{26} = (l * q_3) * q_3 = 3 * 2 = 3;$	**33**		
$v_{27} = l * q_3 = 3 * 2 = 3,$ $v_{28} = (l * q_3) * q_4 = 3 * 3 = 0;$	**30**		
$v_{29} = l * q_3 = 3 * 2 = 3,$ $v_{30} = (l * q_3) * q_5 = 3 * 0 = 2;$	32 (лишний)		

Представленная атака требует 50 операций “$*$”. Таблица Кэли в этой атаке выводится полностью.

2) Можно построить более короткий зашифрованный текст, состоящий из 32 символов (усеченная атака). Эти символы в предыдущей атаке выделены жирным шрифтом и их достаточно для полного восстановления таблицы Кэли операции “$*$”.

3) Перейдем к потоковой атаке Войводы. Вводим следующий текст в устройство шифрования:

$$q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5 \quad q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5 \quad q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5 \quad q_4q_1q_4q_2 \quad \text{или} \quad 0001020304 \quad 1011121314 \quad 2021222324 \quad 3031.$$

Запускаем текст в виде потока. На выходе имеем следующий зашифрованный текст:

Таблица 3.8. Процесс шифрования

$v_1 = l * q_1 = 3 * 0 = 1$	$v_{11} = v_{10} * q_2 = 4 * 1 = 3$	$v_{21} = v_{20} * q_3 = 4 * 2 = 0$
$v_2 = v_1 * q_1 = 1 * 0 = 0$	$v_{12} = v_{11} * q_1 = 3 * 0 = 1$	$v_{22} = v_{21} * q_1 = 0 * 0 = 2$
$v_3 = v_2 * q_1 = 0 * 0 = 2$	$v_{13} = v_{12} * q_2 = 1 * 1 = 1$	$v_{23} = v_{22} * q_3 = 2 * 2 = 4$
$v_4 = v_3 * q_2 = 2 * 1 = 2$	$v_{14} = v_{13} * q_2 = 1 * 1 = 1$	$v_{24} = v_{23} * q_2 = 4 * 1 = 3$
$v_5 = v_4 * q_1 = 2 * 0 = 3$	$v_{15} = v_{14} * q_2 = 1 * 1 = 1$	$v_{25} = v_{24} * q_3 = 3 * 2 = 3$
$v_6 = v_5 * q_3 = 3 * 2 = 3$	$v_{16} = v_{15} * q_3 = 1 * 2 = 2$	$v_{26} = v_{25} * q_3 = 3 * 2 = 3$
$v_7 = v_6 * q_1 = 3 * 0 = 1$	$v_{17} = v_{16} * q_2 = 2 * 1 = 2$	$v_{27} = v_{26} * q_3 = 3 * 2 = 3$
$v_8 = v_7 * q_4 = 1 * 3 = 3$	$v_{18} = v_{17} * q_4 = 2 * 3 = 1$	$v_{28} = v_{27} * q_4 = 3 * 3 = 0$
$v_9 = v_8 * q_1 = 3 * 0 = 1$	$v_{19} = v_{18} * q_2 = 1 * 1 = 1$	$v_{29} = v_{28} * q_3 = 0 * 2 = 1$
$v_{10} = v_9 * q_5 = 1 * 4 = 4$	$v_{20} = v_{19} * q_5 = 1 * 4 = 4$	$v_{30} = v_{29} * q_5 = 1 * 4 = 4$

		$v_{31} = v_{30} * q_4 = 4 * 3 = 2$
		$v_{32} = v_{31} * q_1 = 2 * 0 = 3$
		$v_{33} = v_{32} * q_4 = 3 * 3 = 0$
		$v_{34} = v_{33} * q_2 = 0 * 1 = 0$

Этих символов достаточно для полного восстановления таблицы Кэли операции « $*$ », определенной на Q. В нашем примере вместо 50 символов используется только 34 символа. Количество используемых символов зависит от значения лидера. Получаем для $l = 0$ и $l = 4$ необходимо 30 символов, а для $l = 1$ и $l = 2$ требуется 31 символ.

4) Рассмотрим другой вариант потоковой атаки на открытый текст:

$$q_1q_1q_1q_1q_2q_2q_3q_4q_2q_3 \quad \text{или} \quad 0000112312$$
$$q_2q_3q_3q_4q_4q_4 \qquad\qquad\quad 122333.$$

На выходе имеем следующий зашифрованный текст:

Таблица 3.9. Процесс шифрования

$v_1 = l * q_1 = 3 * 0 = 1$	1	$v_9 = v_8 * q_2 = 0 * 1 = 0$	0
$v_2 = v_1 * q_1 = 1 * 0 = 0$	0	$v_{10} = v_9 * q_3 = 0 * 2 = 1$	1
$v_3 = v_2 * q_1 = 0 * 0 = 2$	2	$v_{11} = v_{10} * q_2 = 1 * 1 = 1$	1
$v_4 = v_3 * q_1 = 2 * 0 = 3$	3	$v_{12} = v_{11} * q_3 = 1 * 2 = 2$	2
$v_5 = v_4 * q_2 = 3 * 1 = 4$	4	$v_{13} = v_{12} * q_3 = 2 * 2 = 4$	4
$v_6 = v_5 * q_2 = 4 * 1 = 3$	3	$v_{14} = v_{13} * q_4 = 4 * 3 = 2$	2
$v_7 = v_6 * q_3 = 3 * 2 = 3$	3	$v_{15} = v_{14} * q_4 = 2 * 3 = 1$	1
$v_8 = v_7 * q_4 = 3 * 3 = 0$	0	$v_{16} = v_{15} * q_4 = 1 * 3 = 3$	3
		$v_{17} = v_{16} * q_4 = 3 * 4 = 2$	2

Представленная атака требует операций “ $*$ ” меньше, чем в атаке, предложенной М. Войводой, но все зависит от выбранного лидера. В нашем примере вместо 50 символов используется только 17. Это минимальный текст из всех возможных в данном примере.

5) Рассмотрим модифицированную атаку с использованием выбранного открытого текста. Вводим следующий текст в устройство шифрования:

$$q_1q_1\ q_2q_2\ q_3q_3\ \ q_4q_4\ q_5q_5 \qquad\qquad 00\ 11\ 22\ 33\ 44$$
$$q_2q_1\ q_3q_2\ q_4q_3\ \ q_5q_4\ q_1q_5 \quad \text{или} \quad 10\ 21\ 32\ 43\ 04$$
$$q_3q_1\ q_4q_2\ q_5q_3\ \ q_1q_4\ q_2q_5 \qquad\qquad 20\ 31\ 42\ 03\ 14$$
$$q_4q_1 \qquad\qquad\qquad\qquad\qquad\qquad 30.$$

На выходе имеем следующий зашифрованный текст:

Таблица 3.10. Процесс шифрования

$v_1 = l * q_1 = 3 * 0 = 1, v_2 = (l * q_1) * q_1 = 1 * 0 = 0;$	10
$v_3 = l * q_2 = 3 * 1 = 4,\ v_4 = (l * q_2) * q_2 = 4 * 1 = 3;$	43
$v_5 = l * q_3 = 3 * 2 = 3,\ v_6 = (l * q_3) * q_3 = 3 * 2 = 3;$	33
$v_7 = l * q_4 = 3 * 3 = 0,\ v_8 = (l * q_4) * q_4 = 0 * 3 = 4;$	04
$v_9 = l * q_5 = 3 * 4 = 2,\ v_{10} = (l * q_5) * q_5 = 2 * 4 = 0;$	20
$v_{11} = l * q_2 = 3 * 1 = 4, v_{12} = (l * q_2) * q_1 = 4 * 0 = 4;$	44
$v_{13} = l * q_3 = 3 * 2 = 3,\ v_{14} = (l * q_3) * q_2 = 3 * 1 = 4;$	34
$v_{15} = l * q_4 = 3 * 3 = 0, v_{16} = (l * q_4) * q_3 = 0 * 2 = 1;$	01
$v_{17} = l * q_5 = 3 * 4 = 2,\ v_{18} = (l * q_5) * q_4 = 2 * 3 = 1;$	21
$v_{19} = l * q_1 = 3 * 0 = 1,\ v_{20} = (l * q_1) * q_5 = 1 * 4 = 4;$	14
$v_{21} = l * q_3 = 3 * 2 = 3, v_{22} = (l * q_3) * q_1 = 3 * 0 = 1;$	31
$v_{23} = l * q_4 = 3 * 3 = 0, v_{23} = (l * q_4) * q_2 = 0 * 1 = 0;$	00
$v_{25} = l * q_5 = 3 * 4 = 2, v_{26} = (l * q_5) * q_3 = 2 * 2 = 4;$	24
$v_{27} = l * q_1 = 3 * 0 = 1,\ v_{28} = (l * q_1) * q_4 = 1 * 3 = 3;$	13
$v_{29} = l * q_2 = 3 * 1 = 4,\ v_{30} = (l * q_2) * q_5 = 4 * 4 = 1;$	41
$v_{31} = l * q_4 = 3 * 3 = 0, v_{32} = (l * q_4) * q_1 = 0 * 0 = 2.$	02

В этом примере вместо 50 символов используются только 32 символа.

Последняя атака дает тот же результат, что и усеченная атака Войводы: 32 символа вместо 50 символов полной атаки Войводы.

Пример 3.2.2. Возьмем квазигруппу $(Q,\ *)$, заданную Таблицей 3.6 из Примера 3.1.4.

Рассмотрим атаки открытым текстом и ограничимся только показом результатов.

1) Введем следующий текст в устройство шифрования:

$$\begin{array}{llllll} \boldsymbol{q_1q_1} & \boldsymbol{q_1q_2} & \boldsymbol{q_1q_3} & \boldsymbol{q_1q_4} & \boldsymbol{q_1q_5} & q_1q_6 \\ \boldsymbol{q_2q_1} & \boldsymbol{q_2q_2} & \boldsymbol{q_2q_3} & \boldsymbol{q_2q_4} & \boldsymbol{q_2q_5} & q_2q_6 \\ \boldsymbol{q_3q_1} & \boldsymbol{q_3q_2} & \boldsymbol{q_3q_3} & \boldsymbol{q_3q_4} & \boldsymbol{q_3q_5} & q_3q_6 \\ \boldsymbol{q_4q_1} & \boldsymbol{q_4q_2} & \boldsymbol{q_4q_3} & \boldsymbol{q_4q_4} & \boldsymbol{q_4q_5} & q_4q_6 \\ \boldsymbol{q_5q_1} & \boldsymbol{q_5q_2} & \boldsymbol{q_5q_3} & \boldsymbol{q_5q_4} & \boldsymbol{q_5q_5} & q_5q_6 \\ q_6q_1 & q_6q_2 & q_6q_3 & q_6q_4 & q_6q_5 & q_6q_6 \end{array} \quad \text{или} \quad \begin{array}{l} \boldsymbol{00\ 01\ 02\ 03\ 04}\ 05 \\ \boldsymbol{10\ 11\ 12\ 13\ 14}\ 15 \\ \boldsymbol{20\ 21\ 22\ 23\ 24}\ 25 \\ \boldsymbol{30\ 31\ 32\ 33\ 34}\ 35 \\ \boldsymbol{40\ 41\ 42\ 43\ 44}\ 45 \\ 50\ 51\ 52\ 53\ 54\ 55. \end{array}$$

На выходе имеем следующий зашифрованный текст:

$$\boldsymbol{22\ 20\ 25\ 23\ 21}\ 24$$

$$\boldsymbol{01\ 03\ 00\ 02\ 04}\ 05$$

55 51 53 54 50 52

30 34 31 35 32 33

13 12 14 11 15 10

44 45 42 40 43 41.

Для полной атаки Войводы требуется 72 символа.

2) Усеченная атака Войводы требует всего 50 символов. Эти символы в предыдущей атаке выделены жирным шрифтом.

3) Теперь вводим в шифровальное устройство следующий текст потоком:

$q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5q_1q_6$ 000102030405

$q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5q_2q_6$ 101112131415

$q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5q_3q_6$ 202122232425

или

$q_4q_1q_4q_2q_4q_3q_4q_4q_4q_5q_4q_6$ 303132333435

$q_5q_1q_5q_2q_5q_3q_5q_4q_5q_5q_5q_6$ 404142434445

$q_6q_1q_6q_2q_6q_3q_6q_4q_6$ 505152535.

На выходе имеем следующий зашифрованный текст:

222014401552

012031234341

442000025005

440353540405

015153232152

4525233.

В этом случае, вместо 72 символов используется только 69 символов (результат зависит от лидера).

4) Рассмотрим другой вариант открытого текста (запускаем текст потоком):

$q_1q_1q_2q_2q_3q_3q_4q_4q_5q_5q_6q_6$ 001122334455

$q_1q_2q_3q_4q_5q_6q_1q_2q_1q_2q_3q_4$ или 012345010123

q_6q_5 54.

На выходе имеем следующий зашифрованный текст:

220314021524

453505513423

32.

Вместо 72 символов можно использовать 26. Это минимальный текст из всех возможных в этом примере.

5) Введем следующий текст в устройство шифрования:

$q_1q_1\ q_2q_2\ q_3q_3\ \ q_4q_4\ q_5q_5\ q_6q_6$		00 11 22 33 44 55
$q_2q_1\ q_3q_2\ q_4q_3\ \ q_5q_4\ q_6q_5\ q_1q_6$		10 21 32 43 54 05
$q_3q_1\ q_4q_2\ q_5q_3\ \ q_6q_4\ q_1q_5\ q_2q_6$	или	20 31 42 53 04 15
$q_4q_1\ q_5q_2\ q_6q_3\ \ q_1q_4\ q_2q_5\ q_3q_6$		30 41 52 03 14 25
q_5q_1		40.

На выходе имеем следующий зашифрованный текст:

22 03 53 35 15 41
01 51 31 11 43 24
55 34 14 40 21 05
30 12 42 23 04 52
13.

В этом случае вместо 72 символов используются только 50.

Даже в бинарном случае при проведении атак выбранным шифротекстом или выбранным открытым текстом количество используемых символов может быть уменьшено.

3.3. Атаки выбранным шифротекстом и выбранным открытым текстом на обобщенный шифр Марковского на основе левых квазигрупп

Посмотрим, как атаки, описанные в предыдущих вопросах, работают в случае использования левой квазигруппы.

Если говорить об усеченной атаке Войводы, то для полной реконструкции таблицы Кэли для левой квазигруппы$(Q,\ \backslash)$ достаточно ввести только $\boldsymbol{2n^2-2n+1}$ символов на входе вместо $\boldsymbol{2n^2}$.

Однако возможна улучшенная версия модифицированной атаки. Запустим следующий текст в декодер:

$$q_1q_1q_2q_2q_3q_3\dots q_{n-2}q_{n-2}q_{n-1}q_{n-1}q_nq_n$$
$$q_2q_1q_3q_2q_4q_3\dots q_{n-1}q_{n-2}q_nq_{n-1}q_1q_n$$
$$q_3q_1q_4q_2q_5q_3\dots q_nq_{n-2}q_1q_{n-1}q_2q_n\dots,$$

устройство дешифрования выдает на выходе следующий открытый текст:

$$l\backslash q_1\ \ q_1\backslash q_1\ q_1\backslash q_2\ \ q_2\backslash q_2\ \dots\ q_n\backslash q_n$$
$$q_n\backslash q_2\ \ q_2\backslash q_1\ \ q_1\backslash q_3\ \ q_3\backslash q_2 \dots q_1\backslash q_n$$
$$q_n\backslash q_3\ \ q_3\backslash q_1\ \ q_1\backslash q_4\ \ q_4\backslash q_2 \dots q_2\backslash q_n \dots .$$

Последний символ зависит от четности порядка квазигруппы, а именно, если n – нечетное число, то последней операцией будет: $q_n\backslash q_k$, где $k = \left[\frac{n}{2}\right] + 1$. Если же n – четное число, то последней операцией будет: $q_n\backslash q_{\frac{n}{2}+1}$.

Представленная атака требует: $\boldsymbol{n^2 - 2\left(n - 1 - \left[\frac{n}{2}\right]\right)}$ операций "\". Если n – нечетное число, то для атаки потребуется: $(n-1)^2 + 2[n/2] + 1$ операций и, если n – четное число, то потребуется: $(n-1)^2 + n + 1$ операций "\". По сравнению с атакой М. Войводы количество используемых символов значительно уменьшено.

Пример 3.3.1. Пусть ключевая левая квазигруппа, с помощью которой производится дешифровка, имеет следующую таблицу Кэли:

Таблица 3.11. Таблица Кэли левой квазигруппы $(\boldsymbol{Q},\ \backslash)$

\	**0**	**1**	**2**	**3**	**4**
0	4	3	2	1	0
1	3	2	1	0	4
2	4	0	3	1	2
3	0	2	1	3	4
4	2	1	0	4	3

Здесь $Q = \{q_1 = 0,\ q_2 = 1,\ q_3 = 2,\ q_4 = 3\}$ и лидер $l = 3$.

1) Введем следующий текст в устройство дешифрования:

$$\begin{array}{lll} \boldsymbol{q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5} & & \boldsymbol{0001020304} \\ \boldsymbol{q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5} & & \boldsymbol{1011121314} \\ \boldsymbol{q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5} & \text{или} & \boldsymbol{2021222324} \\ \boldsymbol{q_4q_1q_4q_2q_4q_3q_4q_4q_4q_5} & & \boldsymbol{3031323334} \\ \boldsymbol{q_5}q_1q_5q_2q_5q_3q_5q_4q_5q_5 & & \boldsymbol{4}041424344. \end{array}$$

На выходе получаем 50 символов. Процесс дешифровки можно посмотреть в Приложении 3, Таблица А3.6. Разбив текст на пять блоков получим:

0443324100

1332210024

0420133112

4012011334

3201402443.

Строки таблицы левой квазигруппы $(Q,\ \backslash)$ выводятся последовательно на четных позициях. Лидер l является решением уравнения: $l \backslash 0 = 0 \Rightarrow l = 3$. Зная таблицу для квазигруппы $(Q,\backslash)$, легко восстанавливается таблица шифрования квазигруппы $(Q,\ *)$:

Таблица 3.12. Таблица Кэли левой квазигруппы $(Q,\ *)$

*	**0**	**1**	**2**	**3**	**4**
0	4	3	2	1	0
1	3	2	1	0	4
2	1	3	4	2	0
3	0	2	1	3	4
4	2	1	0	4	3

2) В этом примере вместо 50 символов будут использованы только первые 41 символов (они выделены жирным шрифтом). Это будет усеченная атака Войводы.

3) Теперь вводим в устройство дешифрования следующий текст (модифицированная атака):

$$q_1q_1q_2q_2q_3q_3q_4q_4q_5q_5 \quad 0011223344$$
$$q_2q_1q_3q_2q_4q_3q_5q_4q_1q_5 \quad \text{или} \quad 1021324304$$
$$q_3 \quad 2.$$

На выходе получаем:

Таблица 3.13. Процесс дешифрования

$u_1 = l\backslash q_1 = 3\backslash 0 = 0$	$u_8 = q_4\backslash q_4 = 3\backslash 3 = 3$	$u_{15} = q_2\backslash q_4 = 1\backslash 3 = 0$
$u_2 = q_1\backslash q_1 = 0\backslash 0 = 4$	$u_9 = q_4\backslash q_5 = 3\backslash 4 = 4$	$u_{16} = q_4\backslash q_3 = 3\backslash 2 = 1$
$u_3 = q_1\backslash q_2 = 0\backslash 1 = 3$	$u_{10} = q_5\backslash q_5 = 4\backslash 4 = 3$	$u_{17} = q_3\backslash q_5 = 2\backslash 4 = 2$
$u_4 = q_2\backslash q_2 = 1\backslash 1 = 2$	$u_{11} = q_5\backslash q_2 = 4\backslash 1 = 1$	$u_{18} = q_5\backslash q_4 = 4\backslash 3 = 4$
$u_5 = q_2\backslash q_3 = 1\backslash 2 = 1$	$u_{12} = q_2\backslash q_1 = 1\backslash 0 = 3$	$u_{19} = q_4\backslash q_1 = 3\backslash 0 = 0$
$u_6 = q_3\backslash q_3 = 2\backslash 2 = 3$	$u_{13} = q_1\backslash q_3 = 0\backslash 2 = 2$	$u_{20} = q_1\backslash q_5 = 0\backslash 4 = 0$
$u_7 = q_3\backslash q_4 = 2\backslash 3 = 1$	$u_{14} = q_3\backslash q_2 = 2\backslash 1 = 0$	$u_{21} = q_5\backslash q_3 = 4\backslash 2 = 0.$

Таким образом, вместо 50 символов будет использоваться только 21 символ.

Пример 3.3.2. Пусть ключевая левая квазигруппа, с помощью которой производится дешифровка, имеет следующую таблицу Кэли:

Таблица 3.14. Таблица Кэли левой квазигруппы $(\boldsymbol{Q},\ \backslash)$

$\backslash$	**0**	**1**	**2**	**3**
0	0	2	1	3
1	1	0	2	3
2	0	3	1	2
3	2	1	3	0

Здесь $Q = \{q_1 = 0,\ q_2 = 1,\ q_3 = 2,\ q_4 = 3\}$ и лидер $l = 2$.

1) Вводим следующий текст в устройство дешифрования:

$$\boldsymbol{q_1q_1q_1q_2q_1q_3q_1q_4} \qquad \boldsymbol{00010203}$$
$$\boldsymbol{q_2q_1q_2q_2q_2q_3q_2q_4} \quad \text{или} \quad \boldsymbol{10111213}$$
$$\boldsymbol{q_3q_1q_3q_2q_3q_3q_3q_4} \qquad \boldsymbol{20212223}$$
$$\boldsymbol{q_4}q_1q_4q_2q_4q_3q_4q_4 \qquad \boldsymbol{3}0313233.$$

Разбив получившийся текст на четыре блока, получим:

00021103

11200233

30132112

02313320.

Для завершения атаки требуется 32 символа. Лидер l является решением уравнения: $l \backslash 0 = 0 \Rightarrow l = 2$. Зная таблицу для квазигруппы $(Q,\ \backslash)$, легко восстанавливается таблица квазигруппы шифрования $(Q,\ *)$:

Таблица 3.15. Таблица Кэли левой квазигруппы $(\boldsymbol{Q},\ *)$

$*$	**0**	**1**	**2**	**3**
0	0	2	1	3
1	1	0	2	3
2	0	2	3	1
3	3	1	0	2

2) В этом примере вместо 32 символов для усеченной атаки будут использоваться только первые 25 символов (они выделены жирным шрифтом).

3) Теперь вводим следующий текст в устройство расшифровки:

$$q_1q_1q_2q_2q_3q_3q_4q_4 \qquad 00112233$$
$$q_2q_1q_3q_2q_4q_3 \quad \text{или} \quad 102132.$$

На выходе получаем:

00202120

111333.

Таким образом, вместо 32 символов будет использоваться только 14 символов.

Для левой квазигруппы модифицированная атака выбранным шифротекстом также дает наилучший результат.

Теперь рассмотрим атаки выбранным открытым текстом, в которых ключевой квазигруппой является левая квазигруппа.

Пример 3.3.3. Пусть ключом шифрования является квазигруппа$(Q, *)$ заданная Таблицей 3.12 из Примера 3.3.1.

1) Введем дискретно по 2 символа следующий текст в устройство шифрования:

$\boldsymbol{q_1q_1}$ $\boldsymbol{q_1q_2}$ $\boldsymbol{q_1q_3}$ $\boldsymbol{q_1q_4}$ q_1q_5		**00 01 02 03** 04
$\boldsymbol{q_2q_1}$ $\boldsymbol{q_2q_2}$ $\boldsymbol{q_2q_3}$ $\boldsymbol{q_2q_4}$ q_2q_5		**10 11 12 13** 14
$\boldsymbol{q_3q_1}$ $\boldsymbol{q_3q_2}$ $\boldsymbol{q_3q_3}$ $\boldsymbol{q_3q_4}$ q_3q_5	или	**20 21 22 23** 24
$\boldsymbol{q_4q_1}$ $\boldsymbol{q_4q_2}$ $\boldsymbol{q_4q_3}$ $\boldsymbol{q_4q_4}$ q_4q_5		**30 31 32 33** 34
$\boldsymbol{q_5q_1}$ $\boldsymbol{q_5q_2}$ $\boldsymbol{q_5q_3}$ $\boldsymbol{q_5q_4}$ q_5q_5		**40 41 42 43** 44.

На выходе имеем следующий зашифрованный текст:

04 03 02 01 00

21 23 24 22 20

13 12 11 10 14

30 32 31 33 34

42 41 40 44 43.

Процесс шифрования можно посмотреть в Приложении 3, Таблица А3.7. На выходе, который идет построчно, на нечетной позиции – номер строки, а на четной позиции – элементы этой строки квазигруппы. Открытый текст состоит из $\boldsymbol{2n^2}$ символов.

2) Можно построить более короткий зашифрованный текст, состоящий из $\boldsymbol{2n^2 - 2n}$ символов (в нашем примере опускаются последние пары, соответствующие элементам последнего столбца, а значит, вместо 50 символов можно использовать только 40). Выход не упорядочен. Это усеченная атака.

3) Если рассматривать атаку следующим открытым текстом (запускается потоком):

$q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5$		0001020304
$q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5$	или	1011121314
$q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5$		2021222324ı

q_4q_1 30.

на выходе имеем следующий зашифрованный текст:

0423021043
2123241034
0403111020
13.

Процесс шифрования можно найти в Приложении 3, Таблица А3.8.

В нашем примере вместо 50 символов запускается только 32, но этот результат изменится при выборе другого лидера и не всегда в лучшую сторону. Для лидера $l = 2$ необходимо 37 контрольных символов, для $l = 0$ и $l = 1$ необходимо 40 символов, а для $l = 4$ требуется 42 символа. Вопрос о пределах изменения числа возможных символов, используемых для восстановления квазигруппы, остается открытым, как и в случае обычной квазигруппы.

4) Теперь рассмотрим другой вариант открытого текста (запускается потоком):

$q_1q_1q_2q_2q_3q_3q_4q_4q_2q_5$ 0011223314
$q_5q_4q_2q_4q_1q_1q_3q_5q_1q_5$ или 4313002404
q_3 2.

На выходе имеем следующий зашифрованный текст:

0412401034
3322131420
2.

Вместо 50 можно использовать 21 символ (Приложение 3, Таблица А3.9.). Это минимальный текст из всех возможных в данном примере.

5) Теперь рассмотрим вариант, когда символы запускаются в шифровальное устройство дискретно, а именно следующими парами:

q_1q_1 q_2q_2 q_3q_3 q_4q_4 q_5q_5 00 11 22 33 44
q_2q_1 q_3q_2 q_4q_3 q_5q_4 q_1q_5 10 21 32 43 04
или
q_3q_1 q_4q_2 q_5q_3 q_1q_4 q_2q_5 20 31 42 03 14
q_4q_1 q_5q_2 q_1q_3 q_2q_4 q_3q_5 30 41 02 13 24.

На выходе имеем следующий зашифрованный текст:

04 23 11 33 43
21 12 31 44 00

13 32 40 01 20

30 41 02 22 14.

Процесс шифрования можно посмотреть в Приложении 3, Таблица А3.10. Вместо 50 символов используется только 40 символов. Этот результат совпадает с результатом усеченной атаки М.Войводы. Открытый текст, используемый в атаке, состоит из $(\boldsymbol{2n^2 - 2n})$ символов, разделенных на пары.

Пример 3.3.4. Возьмем квазигруппу $(Q, *)$ из Примера 3.3.2, заданную Таблицей 3.15, используемой для шифрования. Покажем результаты всех пяти типов атак выбранным шифротекстом.

1) Введем следующий текст в устройство шифрования:

$$\begin{matrix} \boldsymbol{q_1q_1}\ \boldsymbol{q_1q_2}\ \boldsymbol{q_1q_3}\ q_1q_4 \\ \boldsymbol{q}\ \ \boldsymbol{q_1}\ \boldsymbol{q_2q_2}\ \boldsymbol{q_2q_3}\ q_2q_4 \\ \boldsymbol{q_3q_1}\ \boldsymbol{q_3q_2}\ \boldsymbol{q_3q_3}\ q_3q_4 \\ \boldsymbol{q_4q_1}\ \boldsymbol{q_4q_2}\ \boldsymbol{q_4q_3}\ q_4q_4 \end{matrix} \quad \text{или} \quad \begin{matrix} \mathbf{00\ 01\ 02}\ 03 \\ \mathbf{10\ 11\ 12}\ 13 \\ \mathbf{20\ 21\ 22}\ 23 \\ \mathbf{30\ 31\ 32}\ 33. \end{matrix}$$

На выходе имеем следующий зашифрованный текст:

00 02 01 03

20 22 23 21

33 31 30 32

11 10 12 13.

Полная атака Войводы использует 32 символа.

2) Последние пары, соответствующие элементы последнего столбца, можно опустить, а это значит, что вместо 32 символов можно использовать только 24 символа (они выделены жирным шрифтом). Это результат усеченной атаки Войводы.

3) Если рассмотреть атаку следующим открытым текстом:

$$\begin{matrix} q_1q_1q_1q_2q_1q_3q_1q_4 \\ q_2q_1q_2q_2q_2q_3q_2q_4 \\ q_3q_1q_3q_2q_3q_3q_3q_4 \end{matrix} \quad \text{или} \quad \begin{matrix} 00010203 \\ 10111213 \\ 20212223, \end{matrix}$$

на выходе имеем зашифрованный текст:

00020113

11022313

00101232.

В этом случае вместо 32 символов запускается только 24, но этот результат изменится при выборе другого лидера.

4) Рассмотрим другой вариант открытого текста (запускаем текст потоком):

$$\begin{matrix} q_1q_1q_2q_2q_3q_3q_4q_4 \\ q_4q_1q_4q_2q_2 \end{matrix} \text{ или } \begin{matrix} 00112233 \\ 30311. \end{matrix}$$

На выходе имеем следующий зашифрованный текст:

$$00223032$$
$$11310.$$

Вместо 32 символов можно использовать 13 и это минимальный текст из всех возможных в данном примере.

5) Теперь рассмотрим вариант, когда символы запускаются в шифровальное устройство дискретно, а именно следующими парами:

$$\begin{matrix} q_1q_1\ q_2q_2\ q_3q_3\ q_4q_4 \\ q_2q_1\ q_3q_2\ q_4q_3\ q_1q_4 \\ q_3q_1\ q_4q_2\ q_1q_3\ q_2q_4 \end{matrix} \text{ или } \begin{matrix} 00\ 11\ 22\ 33 \\ 10\ 21\ 32\ 03 \\ 20\ 31\ 02\ 13. \end{matrix}$$

На выходе имеем следующий зашифрованный текст:

$$00\ 22\ 30\ 13$$
$$20\ 31\ 12\ 03$$
$$33\ 10\ 01\ 21.$$

В данном примере вместо 32 символов используется только 24 символа.

Наилучший результат для последних двух примеров получен при использовании модифицированной потоковой атаки. В этой атаке количество используемых символов равно $\boldsymbol{(n-1)n+1}$, где $n-$ порядок левой квазигруппы, используемой для шифрования.

3.4. Атаки выбранным шифротекстом и открытым текстом на обобщенный шифр Марковского, основанный на правых квазигруппах

В этом вопросе рассматриваются атаки на текст, построенный на основе правых квазигрупп. Результаты проведенных атак практически совпадают с результатами, полученными для левых квазигрупп, но есть небольшие особенности, на которые следует обратить внимание.

Если мы запустим следующий текст на декодер (модифицированная атака):

$$q_1q_1\ \ q_2q_2\ \ q_3q_3\ \ \cdots\ \ q_{n-2}q_{n-2}\ \ q_{n-1}q_{n-1}\ q_nq_n$$
$$q_2q_1\ \ q_3q_2\ \ q_4q_3\ \ \cdots\ q_{n-1}q_{n-2}\ \ q_nq_{n-1}\ \ q_1q_n$$
$$q_3q_1\ \ q_4q_2\ q_5q_3\ \cdots\ \ q_nq_{n-2}\ \ \ q_1q_{n-1}\ \ q_2\,q_n\ \cdots,$$

устройство дешифрования выдаст на выходе следующий открытый текст:

$$q_1/l, q_1/q_1, q_2/q_1, q_2/q_2, \dots, q_n/q_n,$$
$$q_2/q_n, q_1/q_2, \dots q_n/q_1, q_3/q_n, q_1/q_3,$$
$$q_3/q_n, q_1/q_3, \dots q_n/q_2, q_4/q_n, q_1/q_4, \dots.$$

Последний символ зависит от четности порядка квазигруппы, если $n-$ нечетное число, то последней операцией будет: $q_k\ /q_n$, где $k = \left[\frac{n}{2}\right] + 1$. Если же $n-$ четное число, то последней операцией будет: $q_{\frac{n}{2}+1}/q_n$.

Представленная атака требует: $\boldsymbol{n^2 - 2\left(n - 1 - \left[\frac{n}{2}\right]\right)}$ операций "/".

Пример 3.4.1. Ключевая правая квазигруппа, с помощью которой производится дешифрование, имеет следующую таблицу Кэли:

Таблица 3.16. Таблица Кэли правой квазигруппы $(\boldsymbol{Q},\ /)$

/	**0**	**1**	**2**	**3**	**4**
0	1	2	0	3	4
1	3	4	2	1	0
2	2	1	3	0	2
3	4	3	4	2	1
4	0	0	1	4	3

$Q = \{q_1 = 0,\ q_2 = 1,\ q_3 = 2,\ q_4 = 3,\ q_5 = 4\}$ и лидер $l\ =\ 2$.

Вводим следующий текст в устройство дешифрования:

$$\boldsymbol{q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5} \quad \boldsymbol{0001020304}$$
$$\boldsymbol{q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5} \quad \boldsymbol{1011121314}$$
$$\boldsymbol{q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5} \quad \text{или} \quad \boldsymbol{2021222324}$$
$$\boldsymbol{q_4q_1q_4q_2q_4q_3q_4q_4q_4q_5} \quad \boldsymbol{3031323334}$$
$$\boldsymbol{q_5}q_1q_5q_2q_5q_3q_5q_4q_5q_5 \quad \boldsymbol{4}041424344.$$

На выходе устройства получим:

Таблица 3.17. Процесс дешифрования

$u_1 = q_1/l = 0/2 = 0$	$u_{17} = q_2/q_3 = 1/2 = 2$	$u_{33} = q_4/q_1 = 3/0 = 4$
$u_2 = q_1/q_1 = 0/0 = 1$	$u_{18} = q_4/q_2 = 3/1 = 3$	$u_{34} = q_2/q_4 = 1/3 = 1$
$u_3 = q_1/q_1 = 0/0 = 1$	$u_{19} = q_2/q_4 = 1/3 = 1$	$u_{35} = q_4/q_2 = 3/1 = 3$
$u_4 = q_2/q_1 = 1/0 = 3$	$u_{20} = q_5/q_2 = 4/1 = 0$	$u_{36} = q_3/q_4 = 2/3 = 0$
$u_5 = q_1/q_2 = 0/1 = 2$	$u_{21} = q_3/q_5 = 2/4 = 2$	$u_{37} = q_4/q_3 = 3/2 = 4$
$u_6 = q_3/q_1 = 2/0 = 2$	$u_{22} = q_1/q_3 = 0/2 = 0$	$u_{38} = q_4/q_4 = 3/3 = 2$

$u_7 = q_1/q_3 = 0/2 = 0$	$u_{23} = q_3/q_1 = 2/0 = 2$	$u_{39} = q_4/q_4 = 3/3 = 2$
$u_8 = q_4/q_1 = 3/0 = 4$	$u_{24} = q_2/q_3 = 1/2 = 2$	$u_{40} = q_5/q_4 = 4/3 = 4$
$u_9 = q_1/q_4 = 0/3 = 3$	$u_{25} = q_3/q_2 = 2/1 = 1$	$u_{41} = q_5/q_5 = 4/4 = 3$
$u_{10} = q_5/q_1 = 4/0 = 0$	$u_{26} = q_3/q_3 = 2/2 = 3$	$u_{42} = q_1/q_5 = 0/4 = 4$
$u_{11} = q_2/q_5 = 1/4 = 0$	$u_{27} = q_3/q_3 = 2/2 = 3$	$u_{43} = q_5/q_1 = 4/0 = 0$
$u_{12} = q_1/q_2 = 0/1 = 2$	$u_{28} = q_4/q_3 = 3/2 = 4$	$u_{44} = q_2/q_5 = 1/4 = 0$
$u_{13} = q_2/q_1 = 1/0 = 3$	$u_{29} = q_3/q_4 = 2/3 = 0$	$u_{45} = q_5/q_2 = 4/1 = 0$
$u_{14} = q_2/q_2 = 1/1 = 4$	$u_{30} = q_5/q_3 = 4/2 = 1$	$u_{46} = q_3/q_5 = 2/4 = 2$
$u_{15} = q_2/q_2 = 1/1 = 4$	$u_{31} = q_4/q_5 = 3/4 = 1$	$u_{47} = q_5/q_3 = 4/2 = 1$
$u_{16} = q_3/q_2 = 2/1 = 1$	$u_{32} = q_1/q_4 = 0/3 = 3$	$u_{48} = q_4/q_5 = 3/4 = 1$
		$u_{49} = q_5/q_4 = 4/3 = 4$
		$u_{50} = q_5/q_5 = 4/4 = 3.$

Разбив текст на пять блоков получим:

0113220430

0234412310

2022133401

1341304224

3400021143.

Таким образом, столбцы таблицы правой квазигруппы $(Q, /)$ выводятся последовательно в четных позициях. Лидер l является решением уравнения: $0 / l = 0 \Rightarrow l = 2$. Зная таблицу для квазигруппы $(Q, /)$ легко восстановить таблицу квазигруппы шифрования $(Q, *)$:

Таблица 3.18. Таблица Кэли квазигруппы$(Q, *)$

$*$	**0**	**1**	**2**	**3**	**4**
0	4	4	0	2	1
1	0	2	4	1	3
2	2	0	1	3	2
3	1	3	2	0	4
4	3	1	3	4	0

2) Для полной реконструкции таблицы Кэли для правой квазигруппы $(Q, /)$, как и в случае левой квазигруппы, достаточно ввести только: $\boldsymbol{2n^2 - 2n + 1}$ вместо $\boldsymbol{2n^2}$ символов (в нашем примере вместо 50 символов используется 41символ).

3) Вводим в устройство дешифровки следующий текст:

$q_1q_1q_2q_2q_3q_3q_4q_4q_5q_5$ 0011223344

$q_2q_1q_3q_2q_4q_3q_5q_4q_1q_5$ или 1021324304

q_3 2.

На выходе получаем текст вида:

0134134243

022230113

02.

Так, вместо 50 символов будет использоваться только 21 символ, и результат не зависит от используемого лидера. Процесс дешифрования можно найти в Приложении 3, Таблица А3.11.

Рассмотрим атаки выбранным открытым текстом.

Пример 3.4.2. Пусть ключом шифрования является правая квазигруппа $(Q,*)$, заданная Таблицей 3.18, Пример 3.4.1. Покажем результаты всех пяти типов атак выбранным открытым текстом.

1) Вводим следующий текст в устройство шифрования:

q_1q_1 q_1q_2 q_1q_3 q_1q_4 q_1q_5 00 01 02 03 04

q_2q_1 q_2q_2 q_2q_3 q_2q_4 q_2q_5 10 11 12 13 14

q_3q_1 q_3q_2 q_3q_3 q_3q_4 q_3q_5 или 20 21 22 23 24

q_4q_1 q_4q_2 q_4q_3 q_4q_4 q_4q_5 30 31 32 33 34

q_5q_1 q_5q_2 q_5q_3 q_5q_4 q_5q_5 40 41 42 43 44.

На выходе имеем следующий зашифрованный текст:

Таблица 3.19. Процесс шифрования

$v_1 = q_1 * l = 0 * 2 = 0,$ $v_2 = q_1 * (l * q_1) = 0 * 0 = 4$	04	$v_{27} = q_3 * l = 2 * 2 = 1,$ $v_{28} = q_4 * (l * q_3) = 3 * 1 = 3$	13
$v_3 = q_1 * l = 0 * 2 = 0,$ $v_4 = q_2 * (l * q_1) = 1 * 0 = 0$	00	$v_{29} = q_3 * l = 2 * 2 = 1,$ $v_{30} = q_5 * (l * q_3) = 4 * 1 = 1$	11
$v_5 = q_1 * l = 0 * 2 = 0,$ $v_6 = q_3 * (l * q_1) = 2 * 0 = 2$	02	$v_{31} = q_4 * l = 3 * 2 = 2,$ $v_{32} = q_1 * (l * q_4) = 0 * 2 = 0$	20
$v_7 = q_1 * l = 0 * 2 = 0,$ $v_8 = q_4 * (l * q_1) = 3 * 0 = 1$	01	$v_{33} = q_4 * l = 3 * 2 = 2,$ $v_{34} = q_2 * (l * q_4) = 1 * 2 = 4$	24
$v_9 = q_1 * l = 0 * 2 = 0,$ $v_{10} = q_5 * (l * q_1) = 4 * 0 = 3$	03	$v_{35} = q_4 * l = 3 * 2 = 2,$ $v_{36} = q_3 * (l * q_4) = 2 * 2 = 1$	21

$v_{11} = q_2 * l = 1 * 2 = 4,$ $v_{12} = q_1 * (l * q_2) = 0 * 4 = 1$	41	$v_{37} = q_4 * l = 3 * 2 = 2,$ $v_{38} = q_4 * (l * q_4) = 3 * 2 = 2$	22
$v_{13} = q_2 * l = 1 * 2 = 4,$ $v_{14} = q_2 * (l * q_2) = 1 * 4 = 3$	43	$v_{39} = q_4 * l = 3 * 2 = 2,$ $v_{40} = q_5 * (l * q_4) = 4 * 2 = 3$	23
$v_{15} = q_2 * l = 1 * 2 = 4,$ $v_{16} = q_3 * (l * q_2) = 2 * 4 = 2$	42	$v_{41} = q_5 * l = 4 * 2 = 3,$ $v_{42} = q_1 * (l * q_5) = 0 * 3 = 2$	32
$v_{17} = q_2 * l = 1 * 2 = 4,$ $v_{18} = q_4 * (l * q_2) = 3 * 4 = 4$	44	$v_{43} = q_5 * l = 4 * 2 = 3,$ $v_{44} = q_1 * (l * q_5) = 1 * 3 = 1$	31
$v_{19} = q_2 * l = 1 * 2 = 4,$ $v_{20} = q_5 * (l * q_2) = 4 * 4 = 0$	40	$v_{45} = q_5 * l = 4 * 2 = 3,$ $v_{46} = q_1 * (l * q_5) = 2 * 3 = 3$	33
$v_{21} = q_3 * l = 2 * 2 = 1,$ $v_{22} = q_1 * (l * q_3) = 0 * 1 = 4$	14	$v_{47} = q_5 * l = 4 * 2 = 3,$ $v_{48} = q_1 * (l * q_5) = 3 * 3 = 0$	30
$v_{23} = q_3 * l = 2 * 2 = 1,$ $v_{24} = q_2 * (l * q_3) = 1 * 1 = 2$	12	$v_{49} = q_5 * l = 4 * 2 = 3,$ $v_{50} = q_1 * (l * q_5) = 4 * 3 = 4.$	34
$v_{25} = q_3 * l = 2 * 2 = 1,$ $v_{26} = q_3 * (l * q_3) = 2 * 1 = 0$	10		

Выход идет постолбцовый: на нечетной позиции – номер столбца, а на четной – элемент данного столбца правой квазигруппы $(Q, *)$.

2) Последние пары, соответствующие элементы последней строки, можно опустить, а это значит, что вместо 50 символов можно использовать только 40 символов.

3) Если мы рассмотрим поточную атаку открытым текстом:

$$\begin{array}{lll} q_1q_1q_1q_2q_1q_3q_1q_4q_1q_5 & & 0001020304 \\ q_2q_1q_2q_2q_2q_3q_2q_4q_2q_5 & & 1011121314 \\ q_3q_1q_3q_2q_3q_3q_3q_4q_3q_5 & \text{или} & 2021222324 \\ q_4q_1q_4q_2q_4q_3q_4q_4q_4q_5 & & 3031323334 \\ q_5q_1q_5q_2q_5q_3 & & 404142, \end{array}$$

на выходе получим следующий зашифрованный текст:

$$\begin{array}{l} 0412020140 \\ 0431212240 \\ 2024210103 \\ 0443022223 \\ 411233. \end{array}$$

Процесс шифрования можно найти в Приложении 3, Таблица А3.12. Представленная атака требует обработки 46 элементов, но результат зависит от используемого лидера.

4) Рассмотрим другой вариант открытого текста (осуществляется поточный ввод):

$$q_1q_1q_2q_2q_3q_3q_4q_5q_4q_4 \quad \text{или} \quad 0011223433$$
$$q_4q_2q_2q_4q_1q_4q_1q_3q_1q_5 \quad \text{или} \quad 3113030204$$
$$q_5q_5 \quad \text{или} \quad 44.$$

На выходе имеем следующий зашифрованный текст:

$$0431022301$$
$$2441321403$$
$$34.$$

Вместо 50 можно использовать 21 символ. Это минимальный текст из всех возможных в данном примере. Процесс шифрования можно найти в Приложении 3, Таблица А3.13.

5) Теперь рассмотрим вариант, когда символы запускаются в шифровальное устройство дискретно, а именно следующие пары:

$$q_1q_1\ q_2q_2\ q_3q_3\ q_4q_4\ q_5q_5 \quad \text{или} \quad 00\ 11\ 22\ 33\ 44$$
$$q_2q_1\ q_3q_2\ q_4q_3\ q_5q_4\ q_1q_5 \quad \text{или} \quad 10\ 21\ 32\ 43\ 04$$
$$q_3q_1\ q_4q_2\ q_5q_3\ q_1q_4\ q_2q_5 \quad \text{или} \quad 20\ 31\ 42\ 03\ 14$$
$$q_4q_1\ q_5q_2\ q_1q_3\ q_2q_4\ q_3q_5 \quad \text{или} \quad 30\ 41\ 02\ 13\ 24.$$

На выходе имеем следующий зашифрованный текст:

$$04\ 43\ 10\ 22\ 34$$
$$41\ 12\ 21\ 30\ 03$$
$$14\ 24\ 33\ 01\ 40$$
$$20\ 31\ 02\ 44\ \ 11.$$

Процесс шифрования можно посмотреть в Приложении 3, Таблица А3.14. В этом случае вместо 50 символов используется только 40 символов.

Рассмотренные примеры подтверждают наши выводы об усеченных атаках и модифицированных атаках. Таким образом, криптоаналитику предоставлена возможность выбрать наиболее удобный для него тип атаки.

3.5. Выводы по Главе 3

В третьей главе описаны атаки с использованием выбранного шифротекста и выбранного открытого текста на шифры, полученные с помощью классического алгоритма Марковского и обобщенных алгоритмов Марковского для бинарных квазигрупп. Даже в бинарном случае количество используемых символов может быть уменьшено. Был сделан ряд важных оценок по всем проведенным типам атак.

На основании исследования, проведенного в Главе 3 и полученных результатов, можно сделать следующие выводы:

1) Атаки выбранным шифротекстом и выбранным открытым текстом проводились на шифрах Марковского, построенных с использованием квазигрупп, левых квазигрупп и правых квазигрупп [187];
2) Рассмотрены модификации построенных М. Войводой криптографических атак для квазигрупп, проведен сравнительный анализ (с помощью предельных переходов), выявлены положительные и отрицательные стороны этих атак и предложены новые модифицированные атаки с улучшенными результатами [188];
3) Были отобраны тексты минимальной длины для каждой атаки. Результаты отображаются в Таблице 3.20 [189, 190];
4) Для потоковых атак выбранным открытым текстом определено минимальное необходимое количество символов для полного восстановления таблицы квазигруппы шифрования: $\boldsymbol{2n^2 - 2n + 2}$. Текст зависит от используемого лидера и определяется индивидуально для каждого случая [188-190].

В этой главе решаются задачи, связанные с криптоанализом шифров, построенных во второй главе на основе бинарных квазигрупп.

Результаты, представленные в Главе 3, были опубликованы в [187-190].

Таблица 3.20. Результаты бинарных атак

Порядок	**Необходимое количество используемых символов**			
	Атака М. Войводы выбранным шифротекстом и открытым текстом	Атака М. Войводы выбранным шифротекстом (усеченная)	Атака модифицированным шифротекстом	Атака М. Войводы выбранным открытым текстом (усеченная)
КВАЗИГРУППА				
$\boldsymbol{n}$	$\boldsymbol{2n^2}$	$\boldsymbol{2n^2-4n+1}$	$\boldsymbol{n^2-2(n-1)}$	$\boldsymbol{2(n-1)^2}$
$n=3$	18	7	5	8
$n=4$	32	17	10	18
$n=5$	50	31	17	32
$n=6$	72	49	26	50
$n=7$	98	71	37	72
$n=8$	128	97	50	98
$n=9$	162	127	65	128
$n=10$	200	161	82	162
$n=128$	32768	32257	16130	32258
$n=256$	131072	130049	65026	130050
$n=512$	524288	522241	261122	522242
$n=1024$	2097152	2093057	1046530	2093058
ЛЕВАЯ И ПРАВАЯ КВАЗГРУППА				
$\boldsymbol{n}$	$\boldsymbol{2n^2}$	$\boldsymbol{2n^2-2n+1}$	$\boldsymbol{n^2-2\left(n-1-\left[\frac{n}{2}\right]\right)}$	$\boldsymbol{2n^2-2n}$
$n=3$	18	13	5	7
$n=4$	32	25	10	14
$n=5$	50	41	17	21
$n=6$	72	61	26	32
$n=7$	98	85	37	43
$n=8$	128	113	50	58
$n=9$	162	145	65	73
$n=10$	200	181	82	92
$n=128$	32768	32513	16258	32512
$n=256$	131072	130561	65282	130560
$n=512$	524288	523265	261634	523264
$n=1024$	2097152	2095105	1047554	2095104

4. КРИПТОАНАЛИЗ ПОТОКОВЫХ ШИФРОВ (n-АРНЫЙ СЛУЧАЙ)

В этой главе проводится криптоанализ шифров, построенных с использованием обобщенного алгоритма Марковского, основанного на i-обратимых n-арных группоидах, а именно криптоанализ шифров, основанных на Обобщенном Алгоритме 1.

Первоначальная задача таких атак – взломать ключ, т.е. таблицу значений шифрующей или дешифрующей функции. Даются нижние границы для криптографических атак, называемых атаками выбранным зашифрованным текстом и выбранным открытым текстом. Приведены различные модификации этих атак.

4.1. Атаки выбранным шифротекстом, построенным на основе i-обратимого n – арного группоида

Рассмотрим атаку текстом, построенным с использованием n-арного группоида, который обратим на i-м месте, полученного с помощью обобщенного алгоритма Марковского.

Предположим, что у криптоаналитика есть доступ к устройству дешифрования, загруженному ключом. Затем он может построить следующий зашифрованный текст, где n – арность, а m – порядок i-обратимого группоида:

$$\underbrace{q_1q_1\cdots q_1q_1}_{n \text{ раз}}\underbrace{q_1q_1\cdots q_1q_2}\cdots\underbrace{q_1q_m\cdots q_mq_m}$$

$$\underbrace{q_2q_1\cdots q_1q_1}\underbrace{q_2q_1\cdots q_1q_2}\cdots\underbrace{q_2q_m\cdots q_mq_m}$$

$$\underbrace{q_3q_1\cdots q_1q_1}\underbrace{q_3q_1\cdots q_1q_2}\cdots\underbrace{q_3q_m\cdots q_mq_m}\cdots,$$

и ввести его в дешифрующее устройство. Этот текст является обобщенной версией текста, использованного М. Войводой для бинарных квазигрупп.

Для полного восстановления таблицы значений операции ${}^{(i,n+1)}f$, а значит и таблицы значений операции f, достаточно подать на вход $(\boldsymbol{n\cdot m^{n-1}+1})(\boldsymbol{m-1})$ символов, чтобы получить все значения. Удалось определить минимальную длину необходимого текста для проведения успешной атаки и для ряда случаев построить такие тексты.

Пример 4.1.1. Возьмем тернарный группоид из Примера 2.4.6. Тернарная операция f и обратная операция для f определялись так:

$f(x_1,x_2,x_3)=\alpha x_1+\beta x_2+\gamma x_3=x_4,$

${}^{(3,4)}f(x_1,x_2,x_4)=x_3=\gamma^{-1}(2\cdot\alpha x_1+2\cdot\beta x_2+x_4)$, где

$\alpha 0=1,\ \alpha 1=1,\ \alpha 2=0,$

$\beta 0 = 1,\ \beta 1 = 1,\ \beta 2 = 2,$

$\gamma 0 = 1,\ \gamma 1 = 2,\ \gamma 2 = 0,$

$\gamma^{-1}(0) = 2,\ \gamma^{-1}(1) = 0,\ \gamma^{-1}(2) = 1$.

Выбраны следующие значения лидеров: $l_1 = 1, l_2 = 2, l_3 = 0, l_4 = 1.$

В этом примере использовался Обобщенный Алгоритм 1 (Алгоритм 2.4.5) для шифрования и дешифрования.

Отправляем следующий текст на устройство дешифрования:

$$00000100201001101202002102 2$$

$$10010110211011111212012112220.$$

Процесс дешифровки представлен в следующей таблице:

Таблица 4.1. Процесс дешифрования

$u_1 = {}^{(3,4)}f(l_1, l_2, q_1) = {}^{(3,4)}f(1,2,0) = 2$	$u_{29} = {}^{(3,4)}f(q_3, q_2, q_1) = {}^{(3,4)}f(2,1,0) = 1$
$u_2 = {}^{(3,4)}f(l_3, l_4, q_1) = {}^{(3,4)}f(0,1,0) = 0$	$u_{30} = {}^{(3,4)}f(q_2, q_1, q_1) = {}^{(3,4)}f(1,0,0) = 0$
$u_3 = {}^{(3,4)}f(q_1, q_1, q_1) = {}^{(3,4)}f(0,0,0) = 0$ **−(1)**	$u_{31} = {}^{(3,4)}f(q_1, q_1, q_2) = {}^{(3,4)}f(0,0,1) = 1$
$u_4 = {}^{(3,4)}f(q_1, q_1, q_1) = {}^{(3,4)}f(0,0,0) = 0$	$u_{32} = {}^{(3,4)}f(q_1, q_2, q_1) = {}^{(3,4)}f(0,1,0) = 0$
$u_5 = {}^{(3,4)}f(q_1, q_1, q_1) = {}^{(3,4)}f(0,0,0) = 0$	$u_{33} = {}^{(3,4)}f(q_2, q_1, q_2) = {}^{(3,4)}f(1,0,1) = 1$
$u_6 = {}^{(3,4)}f(q_1, q_1, q_2) = {}^{(3,4)}f(0,0,1) = 1$ **−(2)**	$u_{34} = {}^{(3,4)}f(q_1, q_2, q_2) = {}^{(3,4)}f(0,1,1) = 1$
$u_7 = {}^{(3,4)}f(q_1, q_2, q_1) = {}^{(3,4)}f(0,1,0) = 0$ **−(4)**	$u_{35} = {}^{(3,4)}f(q_2, q_2, q_1) = {}^{(3,4)}f(1,1,0) = 0$
$u_8 = {}^{(3,4)}f(q_2, q_1, q_1) = {}^{(3,4)}f(1,0,0) = 0$ **−(10)**	$u_{36} = {}^{(3,4)}f(q_2, q_1, q_3) = {}^{(3,4)}f(1,0,2) = 2$
$u_9 = {}^{(3,4)}f(q_1, q_1, q_3) = {}^{(3,4)}f(0,0,2) = 2$ **−(3)**	$u_{37} = {}^{(3,4)}f(q_1, q_3, q_2) = {}^{(3,4)}f(0,2,1) = 0$
$u_{10} = {}^{(3,4)}f(q_1, q_3, q_1) = {}^{(3,4)}f(0,2,0) = 2$ **−(7)**	$u_{38} = {}^{(3,4)}f(q_3, q_2, q_2) = {}^{(3,4)}f(2,1,1) = 2$ **−(23)**
$u_{11} = {}^{(3,4)}f(q_3, q_1, q_2) = {}^{(3,4)}f(2,0,1) = 2$ **−(20)**	$u_{39} = {}^{(3,4)}f(q_2, q_2, q_1) = {}^{(3,4)}f(1,1,0) = 0$
$u_{12} = {}^{(3,4)}f(q_1, q_2, q_1) = {}^{(3,4)}f(0,1,0) = 0$	$u_{40} = {}^{(3,4)}f(q_2, q_1, q_2) = {}^{(3,4)}f(1,0,1) = 1$
$u_{13} = {}^{(3,4)}f(q_2, q_1, q_1) = {}^{(3,4)}f(1,0,0) = 0$	$u_{41} = {}^{(3,4)}f(q_1, q_2, q_2) = {}^{(3,4)}f(0,1,1) = 1$
$u_{14} = {}^{(3,4)}f(q_1, q_1, q_2) = {}^{(3,4)}f(0,0,1) = 1$	$u_{42} = {}^{(3,4)}f(q_2, q_2, q_2) = {}^{(3,4)}f(1,1,1) = 1$ **−(14)**
$u_{15} = {}^{(3,4)}f(q_1, q_2, q_2) = {}^{(3,4)}f(0,1,1) = 1$ **−(5)**	$u_{43} = {}^{(3,4)}f(q_2, q_2, q_2) = {}^{(3,4)}f(1,1,1) = 1$
$u_{16} = {}^{(3,4)}f(q_2, q_2, q_1) = {}^{(3,4)}f(1,1,0) = 0$ **−(13)**	$u_{44} = {}^{(3,4)}f(q_2, q_2, q_2) = {}^{(3,4)}f(1,1,1) = 1$
$u_{17} = {}^{(3,4)}f(q_2, q_1, q_2) = {}^{(3,4)}f(1,0,1) = 1$ **−(11)**	$u_{45} = {}^{(3,4)}f(q_2, q_2, q_3) = {}^{(3,4)}f(1,1,2) = 2$ **−(15)**
$u_{18} = {}^{(3,4)}f(q_1, q_2, q_3) = {}^{(3,4)}f(0,1,2) = 2$ **−(6)**	$u_{46} = {}^{(3,4)}f(q_2, q_3, q_2) = {}^{(3,4)}f(1,2,1) = 0$ **−(17)**
$u_{19} = {}^{(3,4)}f(q_2, q_3, q_1) = {}^{(3,4)}f(1,2,0) = 2$ **−(16)**	$u_{47} = {}^{(3,4)}f(q_3, q_2, q_3) = {}^{(3,4)}f(2,1,2) = 0$ **−(24)**
$u_{20} = {}^{(3,4)}f(q_3, q_1, q_3) = {}^{(3,4)}f(2,0,2) = 0$ **−(21)**	$u_{48} = {}^{(3,4)}f(q_2, q_3, q_1) = {}^{(3,4)}f(1,2,0) = 2$
$u_{21} = {}^{(3,4)}f(q_1, q_3, q_1) = {}^{(3,4)}f(0,2,0) = 2$	$u_{49} = {}^{(3,4)}f(q_3, q_1, q_2) = {}^{(3,4)}f(2,0,1) = 2$

$u_{22} = {}^{(3,4)}f(q_3, q_1, q_1) = {}^{(3,4)}f(2,0,0) = 1$ **−(19)**	$u_{50} = {}^{(3,4)}f(q_1, q_2, q_3) = {}^{(3,4)}f(0,1,2) = 2$
$u_{23} = {}^{(3,4)}f(q_1, q_1, q_3) = {}^{(3,4)}f(0,0,2) = 2$	$u_{51} = {}^{(3,4)}f(q_2, q_3, q_2) = {}^{(3,4)}f(1,2,1) = 0$
$u_{24} = {}^{(3,4)}f(q_1, q_3, q_2) = {}^{(3,4)}f(0,2,1) = 0$ **−(8)**	$u_{52} = {}^{(3,4)}f(q_3, q_2, q_2) = {}^{(3,4)}f(2,1,1) = 2$
$u_{25} = {}^{(3,4)}f(q_3, q_2, q_1) = {}^{(3,4)}f(2,1,0) = 1$ **−(22)**	$u_{53} = {}^{(3,4)}f(q_2, q_2, q_3) = {}^{(3,4)}f(1,1,2) = 2$
$u_{26} = {}^{(3,4)}f(q_2, q_1, q_3) = {}^{(3,4)}f(1,0,2) = 2$ **−(12)**	$u_{54} = {}^{(3,4)}f(q_2, q_3, q_3) = {}^{(3,4)}f(1,2,2) = 1$ **−(18)**
$u_{27} = {}^{(3,4)}f(q_1, q_3, q_3) = {}^{(3,4)}f(0,2,2) = 1$ **−(9)**	$u_{55} = {}^{(3,4)}f(q_3, q_3, q_3) = {}^{(3,4)}f(2,2,2) = 2$ **−(27)**
$u_{28} = {}^{(3,4)}f(q_3, q_3, q_2) = {}^{(3,4)}f(2,2,1) = 1$ **−(26)**	$u_{56} = {}^{(3,4)}f(q_3, q_3, q_1) = {}^{(3,4)}f(2,2,0) = 0$ **−(25)**

На выходе этого устройства получаем следующие 56 символов:

$$20000100222001101220212012111010110202011111200222022120.$$

Для полной реконструкции таблицы значений операции ${}^{(3,4)}f$ достаточно подать на вход 56 символов для нашего группоида. В результате взломана таблица дешифрующей функции – Таблица 2.9 из Примера 2.4.6. Зная все значения для операции ${}^{(3,4)}f$, легко восстановить все значения для операции f (Таблица 2.8, Пример 2.4.6).

Чтобы понять ситуацию со взломом дешифрованного текста и лидеров, рассмотрим открытый текст вида: $201121 = u_1u_2u_3u_4u_5u_6$. Попробуем восстановить из этого текста зашифрованный текст, запущенный на дешифратор:

$v_1 = f(l_1, l_2, u_1) = f(l_1, l_2, 2) =?,$

$v_2 = f(l_3, l_4, u_2) = f(l_3, l_4, 0) =?,$

$v_3 = f(v_1, v_2, u_3) = f(v_1, v_2, 1) =?,$

$v_4 = f(v_2, v_3, u_4) = f(v_2, v_3, 1) =?,$

$v_5 = f(v_3, v_4, u_5) = f(v_3, v_4, 2) =?,$

$v_6 = f(v_4, v_5, u_6) = f(v_4, v_5, 1) =?.$

Анализируя результаты, полученные с помощью таблицы значений функции f (Таблица 2.8, Пример 2.4.6), имеем: $f(*,*,2)$, $f(*,*,0)$ и $f(*,*,1)$ могут принимать любые значения. Получаем следующие варианты дешифрованного текста:

Таблица 4.2. Возможные значения дешифрованного текста

v_1	v_2	v_3	v_4	v_5	v_6
0	0	$f(v_1, v_2, 1) =$ $= f(0,0,1) = 1$	$f(v_2, v_3, 1) =$ $= f(0,1,1) = 1$	$f(v_3, v_4, 2) =$ $= f(1,1,2) = 2$	$f(v_4, v_5, 1) =$ $= f(1,2,1) = 2$
0	1	$f(v_1, v_2, 1) =$ $= f(0,1,1) = 1$	$f(v_2, v_3, 1) =$ $= f(1,1,1) = 1$	$f(v_3, v_4, 2) =$ $= f(1,1,2) = 2$	$f(v_4, v_5, 1) =$ $= f(1,2,1) = 2$

0	2	$f(v_1, v_2, 1) =$ $= f(0,2,1) = 2$	$f(v_2, v_3, 1) =$ $= f(2,2,1) = 1$	$f(v_3, v_4, 2) =$ $= f(2,1,2) = 1$	$f(v_4, v_5, 1) =$ $= f(1,1,1) = 1$
1	0	$f(v_1, v_2, 1) =$ $= f(1,0,1) = 1$	$f(v_2, v_3, 1) =$ $= f(0,1,1) = 1$	$f(v_3, v_4, 2) =$ $= f(1,1,2) = 2$	$f(v_4, v_5, 1) =$ $= f(1,2,1) = 2$
1	1	$f(v_1, v_2, 1) =$ $= f(1,1,1) = 1$	$f(v_2, v_3, 1) =$ $= f(1,1,1) = 1$	$f(v_3, v_4, 2) =$ $= f(1,1,2) = 2$	$f(v_4, v_5, 1) =$ $= f(1,2,1) = 2$
1	2	$f(v_1, v_2, 1) =$ $= f(1,2,1) = 2$	$f(v_2, v_3, 1) =$ $= f(2,2,1) = 1$	$f(v_3, v_4, 2) =$ $= f(2,1,2) = 1$	$f(v_4, v_5, 1) =$ $= f(1,1,1) = 1$
2	0	$f(v_1, v_2, 1) =$ $= f(2,0,1) = 0$	$f(v_2, v_3, 1) =$ $= f(0,0,1) = 1$	$f(v_3, v_4, 2) =$ $= f(0,1,2) = 2$	$f(v_4, v_5, 1) =$ $= f(1,2,1) = 2$
2	1	$f(v_1, v_2, 1) =$ $= f(2,1,1) = 0$	$f(v_2, v_3, 1) =$ $= f(1,0,1) = 1$	$f(v_3, v_4, 2) =$ $= f(0,1,2) = 2$	$f(v_4, v_5, 1) =$ $= f(1,2,1) = 2$
2	2	$f(v_1, v_2, 1) =$ $= f(2,2,1) = 1$	$f(v_2, v_3, 1) =$ $= f(2,1,1) = 0$	$f(v_3, v_4, 2) =$ $= f(1,0,2) = 2$	$f(v_4, v_5, 1) =$ $= f(0,2,1) = 2.$

Получаем 9 вариантов возможного дешифрованного текста:

№	v_1	v_2	v_3	v_4	v_5	v_6
(1)	**0**	**0**	**1**	**1**	**2**	**2**
(2)	0	1	1	1	2	2
(3)	0	2	2	1	1	1
(4)	1	0	1	1	2	2
(5)	1	1	1	1	2	2
(6)	1	2	2	1	1	1
(7)	2	0	0	1	2	2
(8)	2	1	0	1	2	2
(9)	2	2	1	0	2	2

Среди которых первый вариант верный. Возможных значений шифротекста будет всего 9 вариантов, т.е. определить истинное значение не представляет особой сложности.

Ситуация с лидерами в примере следующая: $f(l_1, l_2, 2) = f(*,*,2)$ и $f(l_3, l_4, 0) =$ $= f(*,*,0)$, где $f(*,*,2)$ и $f(*,*,0)$ могут принимать любые значения. Вопрос о выявлении лидеров в этом случае теряет свою актуальность. Различных наборов лидеров для тернарного группоида будет: $3^4 = 81$. По сути, точные значения самих лидеров определять не нужно.

Шифротекст, предложенный в Примере 4.1.1, является обобщенной версией шифротекста, использованного М. Войводой для бинарных квазигрупп.

В следующих атаках результат был улучшен. Введем следующий текст в устройство дешифрования:

$$q_1q_1q_1q_2q_2q_2q_3q_3q_3 \quad 000111222$$
$$q_2q_1q_1q_3q_2q_2\ q_1q_3q_3 \quad \text{или} \quad 100211022$$
$$q_1q_2q_1q_2q_3q_2q_3q_1q_3 \quad 010121202$$
$$q_1q_1 \quad 00.$$

Получаем следующий дешифрующий процесс:

Таблица 4.3. Процесс дешифрования

$u_1 = {}^{(3,4)}f(l_1, l_2, q_1) = {}^{(3,4)}f(1,2,0) = 2$
$u_2 = {}^{(3,4)}f(l_3, l_4, q_1) = {}^{(3,4)}f(0,1,0) = 0$
$u_3 = {}^{(3,4)}f(q_1, q_1, q_1) = {}^{(3,4)}f(0,0,0) = 0$ **−(1)**
$u_4 = {}^{(3,4)}f(q_1, q_1, q_2) = {}^{(3,4)}f(0,0,1) = 1$ **−(2)**
$u_5 = {}^{(3,4)}f(q_1, q_2, q_2) = {}^{(3,4)}f(0,1,1) = 1$ **−(5)**
$u_6 = {}^{(3,4)}f(q_2, q_2, q_2) = {}^{(3,4)}f(1,1,1) = 1$ **−(14)**
$u_7 = {}^{(3,4)}f(q_2, q_2, q_3) = {}^{(3,4)}f(1,1,2) = 2$ **−(15)**
$u_8 = {}^{(3,4)}f(q_2, q_3, q_3) = {}^{(3,4)}f(1,2,2) = 1$ **−(18)**
$u_9 = {}^{(3,4)}f(q_3, q_3, q_3) = {}^{(3,4)}f(2,2,2) = 2$ **−(27)**
$u_{10} = {}^{(3,4)}f(q_3, q_3, q_2) = {}^{(3,4)}f(2,2,1) = 1$ **−(26)**
$u_{11} = {}^{(3,4)}f(q_3, q_2, q_1) = {}^{(3,4)}f(2,1,0) = 1$ **−(22)**
$u_{12} = {}^{(3,4)}f(q_2, q_1, q_1) = {}^{(3,4)}f(1,0,0) = 0$ **−(10)**
$u_{13} = {}^{(3,4)}f(q_1, q_1, q_3) = {}^{(3,4)}f(0,0,2) = 2$ **−(3)**
$u_{14} = {}^{(3,4)}f(q_1, q_3, q_2) = {}^{(3,4)}f(0,2,1) = 0$ **−(8)**
$u_{15} = {}^{(3,4)}f(q_3, q_2, q_2) = {}^{(3,4)}f(2,1,1) = 2$ **−(23)**
$u_{16} = {}^{(3,4)}f(q_2, q_2, q_1) = {}^{(3,4)}f(1,1,0) = 0$ **−(13)**
$u_{17} = {}^{(3,4)}f(q_2, q_1, q_3) = {}^{(3,4)}f(1,0,2) = 2 - (\mathbf{12})$
$u_{18} = {}^{(3,4)}f(q_1, q_3, q_3) = {}^{(3,4)}f(0,2,2) = 1 - (\mathbf{9})$
$u_{19} = {}^{(3,4)}f(q_3, q_3, q_1) = {}^{(3,4)}f(2,2,0) = 0 - (\mathbf{25})$
$u_{20} = {}^{(3,4)}f(q_3, q_1, q_2) = {}^{(3,4)}f(2,0,1) = 2 - (\mathbf{20})$
$u_{21} = {}^{(3,4)}f(q_1, q_2, q_1) = {}^{(3,4)}f(0,1,0) = 0$ **−(4)**
$u_{22} = {}^{(3,4)}f(q_2, q_1, q_2) = {}^{(3,4)}f(1,0,1) = 1 - (\mathbf{11})$
$u_{23} = {}^{(3,4)}f(q_1, q_2, q_3) = {}^{(3,4)}f(0,1,2) = 2$ **−(6)**

$u_{24} = {}^{(3,4)}f(q_2, q_3, q_2) = {}^{(3,4)}f(1,2,1) = 0 - (\mathbf{17})$
$u_{25} = {}^{(3,4)}f(q_3, q_2, q_3) = {}^{(3,4)}f(2,1,2) = 0 -$**(24)**
$u_{26} = {}^{(3,4)}f(q_2, q_3, q_1) = {}^{(3,4)}f(1,2,0) = 2 -$**(16)**
$u_{27} = {}^{(3,4)}f(q_3, q_1, q_3) = {}^{(3,4)}f(2,0,2) = 0 - (\mathbf{21})$
$u_{28} = {}^{(3,4)}f(q_1, q_3, q_1) = {}^{(3,4)}f(0,2,0) = 2 - (\mathbf{7})$
$u_{29} = {}^{(3,4)}f(q_3, q_1, q_1) = {}^{(3,4)}f(2,0,0) = 1 -$**(19)**

На выходе получаем следующие 29 символов:$2001112121102020210201200 2021.$

Таким образом, для полной реконструкции таблицы значений операции ${}^{(3,4)}f$ достаточно подать на вход 29 символов, чтобы восстановить все значения.

Следует отметить, что для значений функции f и обратной ей функции ${}^{(i,n+1)}f$ на следующих множествах: $(q_{j_1}, q_{j_2}, \dots q_{j_{i-1}}, q_i, q_{j_{i+1}}, \dots q_{j_n})$, где элементы $q_{j_1}, q_{j_2}, \dots q_{j_{i-1}}, q_{j_{i+1}}, \dots q_{j_n}$ выбираются из множества $\{q_1, q_2, \dots q_m\}$ и являются фиксированными элементами, при разных значениях элемента $q_i -$ соответствующие функции не могут принимать одинаковые значения. Для каждого такого фиксированного набора достаточно определить $(m-1)$ значение соответствующей функции, и последнее значение будет найдено автоматически. С учетом этого замечания построенный текст будет иметь длину: $\boldsymbol{m^{n-1} \cdot (m-1)}$.

Следует отметить две особенности такого текста:

1) Определение значений остальных функций (их осталось m^{n-1}) является более сложной задачей, чем в случае работы с бинарными квазигруппами;
2) Для случая, когда $n = m = 3$, такой текст найден, но можно ли будет подобрать аналогичный текст в других случаях? И можно ли будет найти общий вид такого текста, или он будет разным для каждого случая? Эти вопросы еще предстоит решить.

Для нашего примера вводится следующий текст в устройство дешифрования:

$q_1q_1q_1q_2q_2q_2q_3q_3q_3$ 000111222
$q_2q_1q_2q_1q_3q_1q_3q_2q_3$ или 101020212
q_1q_2 01.

Получаем следующий дешифрующий процесс:

Таблица 4.4. Процесс дешифрования

$u_1 = {}^{(3,4)}f(l_1, l_2, q_1) = {}^{(3,4)}f(1,2,0) = 2$
$u_2 = {}^{(3,4)}f(l_3, l_4, q_1) = {}^{(3,4)}f(0,1,0) = 0$

$u_3 = {}^{(3,4)}f(q_1, q_1, q_1) = {}^{(3,4)}f(0,0,0) = 0$ **−(1)**
$u_4 = {}^{(3,4)}f(q_1, q_1, q_2) = {}^{(3,4)}f(0,0,1) = 1$ **−(2)**
$u_5 = {}^{(3,4)}f(q_1, q_2, q_2) = {}^{(3,4)}f(0,1,1) = 1$ **−(5)**
$u_6 = {}^{(3,4)}f(q_2, q_2, q_2) = {}^{(3,4)}f(1,1,1) = 1$ **−(14)**
$u_7 = {}^{(3,4)}f(q_2, q_2, q_3) = {}^{(3,4)}f(1,1,2) = 2$ **−(15)**
$u_8 = {}^{(3,4)}f(q_2, q_3, q_3) = {}^{(3,4)}f(1,2,2) = 1$ **−(18)**
$u_9 = {}^{(3,4)}f(q_3, q_3, q_3) = {}^{(3,4)}f(2,2,2) = 2$ **−(27)**
$u_{10} = {}^{(3,4)}f(q_3, q_3, q_2) = {}^{(3,4)}f(2,2,1) = 1$ **−(26)**
$u_{11} = {}^{(3,4)}f(q_3, q_2, q_1) = {}^{(3,4)}f(2,1,0) = 1$ **−(22)**
$u_{12} = {}^{(3,4)}f(q_2, q_1, q_2) = {}^{(3,4)}f(1,0,1) = 1$ **−(11)**
$u_{13} = {}^{(3,4)}f(q_1, q_2, q_1) = {}^{(3,4)}f(0,1,0) = 0$ **−(4)**
$u_{14} = {}^{(3,4)}f(q_2, q_1, q_3) = {}^{(3,4)}f(1,0,2) = 2$ **−(12)**
$u_{15} = {}^{(3,4)}f(q_1, q_3, q_1) = {}^{(3,4)}f(0,2,0) = 2$ **−(7)**
$u_{16} = {}^{(3,4)}f(q_3, q_1, q_3) = {}^{(3,4)}f(2,0,2) = 0$ **−(21)**
$u_{17} = {}^{(3,4)}f(q_1, q_3, q_2) = {}^{(3,4)}f(0,2,1) = 0 - (\mathbf{8})$
$u_{18} = {}^{(3,4)}f(q_3, q_2, q_3) = {}^{(3,4)}f(2,1,2) = 0 - (\mathbf{24})$
$u_{19} = {}^{(3,4)}f(q_2, q_3, q_1) = {}^{(3,4)}f(1,2,0) = 2 - (\mathbf{16})$
$u_{20} = {}^{(3,4)}f(q_3, q_1, q_2) = {}^{(3,4)}f(2,0,1) = 2 - (\mathbf{20})$

На выходе получаем следующие 20 символов: 20011121211102200022.

Таким образом, для восстановления таблицы значений операции ${}^{(3,4)}f$, достаточно ввести 20 символов, чтобы восстановить 18 значений из 27. Этот текст имеет наименьшую длину для случая $n = m = 3$.

Пример 4.1.2. Возьмем 4-арный группоид $(R_3, f), R_3 = \{0,1,2\}$, который определен над кольцом классов вычетов по модулю 3 $-(R_3, +, \cdot)$ и обратим на последнем месте. Мы определим 4-арную операцию f на множестве R_3 следующим образом:

$f(x_1, x_2, x_3, x_4) = \alpha x_1 + \beta x_2 + \gamma x_3 + \delta x_4 = x_5$, где

$\alpha 0 = 1, \quad \alpha 1 = 0, \quad \alpha 2 = 2,$

$\beta 0 = 0, \quad \beta 1 = 0, \quad \beta 2 = 1,$

$\gamma 0 = 2, \quad \gamma 1 = 1, \quad \gamma 2 = 1,$

$\delta 0 = 2, \quad \delta 1 = 0, \quad \delta 2 = 1.$

(45)-парастроф для f имеет вид:

${}^{(4,5)}f(x_1, x_2, x_3, x_5) = x_4 = \delta^{-1}(2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3 + x_5)$, где

$\delta^{-1}(0) = 1, \quad \delta^{-1}(1) = 2, \quad \delta^{-1}(2) = 0.$

Проверим: $f\big(x_1, x_2, x_3, {}^{(4,5)}f(x_1, x_2, x_3, x_5)\big) =$

$= \alpha x_1 + \beta x_2 + \gamma x_3 + \delta(\delta^{-1}(2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3 + x_5)) =$

$= \alpha x_1 + \beta x_2 + \gamma x_3 + 2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3 + x_5 = x_5,$

${}^{(4,5)}f(x_1, x_2, x_3, f(x_1, x_2, x_3, x_4)) =$

$= \delta^{-1}\big(2 \cdot \alpha x_1 + 2 \cdot \beta x_2 + 2 \cdot \gamma x_3 + \alpha x_1 + \beta x_2 + \gamma x_3 + \delta x_4\big) = \delta^{-1}\big(\delta x_4\big) = x_4.$

Лидерами выступают элементы: $l_1 = 1, l_2 = 0,\ l_3 = 0, l_4 = 2, l_5 = 1,\ l_6 = 1,\ l_7 = 0,$ $l_8 = 0,\ l_9 = 0.$

Для шифрования и дешифрования используем Обобщенный Алгоритм 1 (Алгоритм 2.4.5). Введем следующий текст в устройство дешифрования:

00000001000200100011001200200021002 2

01000101010201100111011201200121012 2

02000201020202100211021202200221022 2

10001001100210101011101210201021102 2

11001101110211101111111211201121112 2

12001201120212101211121212201221122 2 20.

В таблице показаны значения символов, которые позволяют определить все нужные значения функции ${}^{(4,5)}f$:

Таблица 4.5. Процесс дешифрования (фрагмент)

$u_4 = {}^{(4,5)}f(0,0,0,0) = 1$ **−(1)**	$u_{69} = {}^{(4,5)}f(1,2,1,0) = 2$ **−(49)**
$u_8 = {}^{(4,5)}f(0,0,0,1) = 2$ **−(2)**	$u_{70} = {}^{(4,5)}f(2,1,0,1) = 1$ **−(65)**
$u_9 = {}^{(4,5)}f(0,0,1,0) = 2$ **−(4)**	$u_{71} = {}^{(4,5)}f(1,0,1,2) = 2$ **−(33)**
$u_{10} = {}^{(4,5)}f(0,1,0,0) = 1$ **−(10)**	$u_{72} = {}^{(4,5)}f(0,1,2,2) = 1$ **−(18)**
$u_{11} = {}^{(4,5)}f(1,0,0,0) = 2$ **−(28)**	$u_{73} = {}^{(4,5)}f(1,2,2,0) = 2$ **−(52)**
$u_{12} = {}^{(4,5)}f(0,0,0,2) = 0$ **−(3)**	$u_{74} = {}^{(4,5)}f(2,2,0,2) = 1$ **−(75)**
$u_{13} = {}^{(4,5)}f(0,0,2,0) = 2$ **−(7)**	$u_{75} = {}^{(4,5)}f(2,0,2,0) = 1$ **−(61)**
$u_{14} = {}^{(4,5)}f(0,2,0,0) = 0$ **−(19)**	$u_{84} = {}^{(4,5)}f(0,2,0,2) = 2$ **−(21)**
$u_{15} = {}^{(4,5)}f(2,0,0,1) = 1$ **−(56)**	$u_{87} = {}^{(4,5)}f(2,0,2,1) = 2$ **−(62)**
$u_{20} = {}^{(4,5)}f(0,0,1,1) = 0$ **−(5)**	$u_{92} = {}^{(4,5)}f(0,2,1,1) = 2$ **−(23)**
$u_{21} = {}^{(4,5)}f(0,1,1,0) = 2$ **−(13)**	$u_{93} = {}^{(4,5)}f(2,1,1,0) = 1$ **−(67)**
$u_{22} = {}^{(4,5)}f(1,1,0,0) = 2$ **−(37)**	$u_{94} = {}^{(4,5)}f(1,1,0,2) = 1$ **−(39)**
$u_{23} = {}^{(4,5)}f(1,0,0,1) = 0$ **−(29)**	$u_{95} = {}^{(4,5)}f(1,0,2,1) = 1$ **−(35)**

$u_{24} = {}^{(4,5)}f(0,0,1,2) = 1$ **−(6)**	$u_{96} = {}^{(4,5)}f(0,2,1,2) = 0$ **−(24)**
$u_{25} = {}^{(4,5)}f(0,1,2,0) = 2$ **−(16)**	$u_{97} = {}^{(4,5)}f(2,1,2,0) = 1$ **−(70)**
$u_{26} = {}^{(4,5)}f(1,2,0,0) = 1$ **−(46)**	$u_{98} = {}^{(4,5)}f(1,2,0,2) = 0$ **−(48)**
$u_{27} = {}^{(4,5)}f(2,0,0,2) = 2$ **−(57)**	$u_{99} = {}^{(4,5)}f(2,0,2,2) = 0$ **−(63)**
$u_{30} = {}^{(4,5)}f(2,0,0,0) = 0$ **−(55)**	$u_{101} = {}^{(4,5)}f(2,2,0,0) = 2$ **−(73)**
$u_{32} = {}^{(4,5)}f(0,0,2,1) = 0$ **−(8)**	$u_{104} = {}^{(4,5)}f(0,2,2,1) = 2$ **−(26)**
$u_{33} = {}^{(4,5)}f(0,2,1,0) = 1$ **−(22)**	$u_{105} = {}^{(4,5)}f(2,2,1,0) = 0$ **−(76)**
$u_{34} = {}^{(4,5)}f(2,1,0,0) = 0$ **−(64)**	$u_{106} = {}^{(4,5)}f(2,1,0,2) = 2$ **−(66)**
$u_{35} = {}^{(4,5)}f(1,0,0,2) = 1$ **−(30)**	$u_{107} = {}^{(4,5)}f(1,0,2,2) = 2$ **−(36)**
$u_{36} = {}^{(4,5)}f(0,0,2,2) = 1$ **−(9)**	$u_{108} = {}^{(4,5)}f(0,2,2,2) = 0$ **−(27)**
$u_{37} = {}^{(4,5)}f(0,2,2,0) = 1$ **−(25)**	$u_{109} = {}^{(4,5)}f(2,2,2,1) = 1$ **−(80)**
$u_{38} = {}^{(4,5)}f(2,2,0,1) = 0$ **−(74)**	$u_{146} = {}^{(4,5)}f(2,2,1,1) = 1$ **−(77)**
$u_{39} = {}^{(4,5)}f(2,0,1,0) = 1$ **−(58)**	$u_{159} = {}^{(4,5)}f(2,1,1,1) = 2$ **−(68)**
$u_{44} = {}^{(4,5)}f(0,1,0,1) = 2$ **−(11)**	$u_{164} = {}^{(4,5)}f(1,1,1,1) = 1 - (\mathbf{41})$
$u_{45} = {}^{(4,5)}f(1,0,1,0) = 0$ **−(31)**	$u_{168} = {}^{(4,5)}f(1,1,1,2) = 2$ **−(42)**
$u_{48} = {}^{(4,5)}f(0,1,0,2) = 0$ **−(12)**	$u_{169} = {}^{(4,5)}f(1,1,2,1) = 1$ **−(44)**
$u_{49} = {}^{(4,5)}f(1,0,2,0) = 0$ **−(34)**	$u_{170} = {}^{(4,5)}f(1,2,1,1) = 0$ **−(50)**
$u_{50} = {}^{(4,5)}f(0,2,0,1) = 1$ **−(20)**	$u_{171} = {}^{(4,5)}f(2,1,1,2) = 0$ **−(69)**
$u_{51} = {}^{(4,5)}f(2,0,1,1) = 2$ **−(59)**	$u_{180} = {}^{(4,5)}f(1,1,2,2) = 2$ **−(45)**
$u_{56} = {}^{(4,5)}f(0,1,1,1) = 0$ **−(14)**	$u_{181} = {}^{(4,5)}f(1,2,2,1) = 0$ **−(53)**
$u_{57} = {}^{(4,5)}f(1,1,1,0) = 0$ **−(40)**	$u_{182} = {}^{(4,5)}f(2,2,1,2) = 2$ **−(78)**
$u_{58} = {}^{(4,5)}f(1,1,0,1) = 0$ **−(38)**	$u_{195} = {}^{(4,5)}f(2,1,2,1) = 2$ **−(71)**
$u_{59} = {}^{(4,5)}f(1,0,1,1) = 1$ **−(32)**	$u_{204} = {}^{(4,5)}f(1,2,1,2) = 1$ **−(51)**
$u_{60} = {}^{(4,5)}f(0,1,1,2) = 1$ **−(15)**	$u_{207} = {}^{(4,5)}f(2,1,2,2) = 0$ **−(72)**
$u_{61} = {}^{(4,5)}f(1,1,2,0) = 0$ **−(43)**	$u_{216} = {}^{(4,5)}f(1,2,2,2) = 1$ **−(54)**
$u_{62} = {}^{(4,5)}f(1,2,0,1) = 2$ **−(47)**	$u_{217} = {}^{(4,5)}f(2,2,2,2) = 2$ **−(81)**
$u_{63} = {}^{(4,5)}f(2,0,1,2) = 0$ **−(60)**	$u_{218} = {}^{(4,5)}f(2,2,2,0) = 0$ **−(79)**
$u_{68} = {}^{(4,5)}f(0,1,2,1) = 0$ **−(17)**	

Полную версию Таблицы 4.5 можно найти в Приложении 3, Таблица А3.15. Так, для полной реконструкции таблицы значений операции ${}^{(4,5)}f$, а значит, и таблицы значений операции f, достаточно ввести 218 символов на входе. Таблицу значений для функции дешифрования также можно найти в Приложении 3, Таблица А3.16. Зная таблицу Кэли для

операции $^{(4,5)}f$, мы легко восстановили таблицу операции f (Приложение 3, Таблица А3.17).

Чтобы разобраться в ситуации со взломом зашифрованного текста и лидеров, рассмотрим открытый текст вида: $101202 = u_1u_2u_3u_4u_5u_6$. Для этого текста имеем:

$v_1 = f(l_1, l_2, l_3, u_1) = f(l_1, l_2, l_3, 1) =?,$

$v_2 = f(l_4, l_5, l_6, u_2) = f(l_4, l_5, l_6, 0) =?,$

$v_3 = f(l_7, l_8, l_8, u_3) = f(l_7, l_8, l_8, 1) =?,$

$v_4 = f(v_1, v_2, v_3, u_4) = f(v_1, v_2, v_3, 2) =?,$

$v_5 = f(v_2, v_3, v_4, u_5) = f(v_2, v_3, v_4, 0) =?,$

$v_6 = f(v_3, v_4, v_5, u_6) = f(v_3, v_4, v_5, 2) =?.$

Анализируя полученные результаты и используя таблицу значений функции f, получаем, что $f(*,*,*,1)$ и $f(*,*,*,0)$ могут принимать любые значения.

Таблица 4.6. Варианты дешифрованного текста

№	v_1	v_2	v_3	v_4	v_5	v_6
(1)	0	0	0	$f(0,0,0,2) = 1$	$f(0,0,1,0) = 1$	$f(0,1,1,2) = 0$
(2)	0	0	1	$f(0,0,1,2) = 0$	$f(0,1,0,0) = 2$	$f(1,0,2,2) = 2$
(3)	0	0	2	$f(0,0,2,2) = 0$	$f(0,2,0,0) = 0$	$f(2,0,0,2) = 2$
(4)	0	1	0	$f(0,1,0,2) = 1$	$f(1,0,1,0) = 0$	$f(0,1,0,2) = 1$
(5)	0	1	1	$f(0,1,1,2) = 0$	$f(1,1,0,0) = 1$	$f(1,0,1,2) = 2$
(6)	0	1	2	$f(0,1,2,2) = 0$	$f(1,2,0,0) = 2$	$f(2,0,2,2) = 1$
(7)	0	2	0	$f(0,2,0,2) = 2$	$f(2,0,2,0) = 2$	$f(0,2,2,2) = 1$
(8)	0	2	1	$f(0,2,1,2) = 1$	$f(2,1,1,0) = 2$	$f(1,1,2,2) = 2$
(9)	0	2	2	$f(0,2,2,2) = 1$	$f(2,2,1,0) = 0$	$f(2,1,0,2) = 2$
(10)	1	0	0	$f(1,0,0,2) = 0$	$f(0,0,0,0) = 2$	$f(0,0,2,2) = 0$
(11)	1	0	1	$f(1,0,1,2) = 2$	$f(0,1,2,0) = 1$	$f(1,2,1,2) = 0$
(12)	1	0	2	$f(1,0,2,2) = 2$	$f(0,2,2,0) = 2$	$f(2,2,2,2) = 2$
(13)	1	1	0	$f(1,1,0,2) = 0$	$f(1,0,0,0) = 1$	$f(0,0,1,2) = 0$
(14)	1	1	1	$f(1,1,1,2) = 2$	$f(1,1,2,0) = 0$	$f(1,2,0,2) = 1$
(15)	1	1	2	$f(1,1,2,2) = 2$	$f(1,2,2,0) = 1$	$f(2,2,1,2) = 2$
(16)	1	2	0	$f(1,2,0,2) = 1$	$f(2,0,1,0) = 2$	$f(0,1,2,2) = 0$
(17)	1	2	1	$f(1,2,1,2) = 0$	$f(2,1,0,0) = 0$	$f(1,0,0,2) = 0$
(18)	1	2	2	$f(1,2,2,2) = 0$	$f(2,2,0,0) = 1$	$f(2,0,1,2) = 1$
(19)	2	0	0	$f(2,0,0,2) = 2$	$f(0,0,2,0) = 1$	$f(0,2,1,2) = 1$

(20)	2	0	1	$f(2,0,1,2)=1$	$f(0,1,1,0)=1$	$f(1,1,1,2)=2$
(21)	2	0	2	$f(2,0,2,2)=1$	$f(0,2,1,0)=2$	$f(2,1,2,2)=1$
(22)	2	1	0	$f(2,1,0,2)=2$	$f(1,0,2,0)=0$	$f(0,2,0,2)=2$
(23)	2	1	1	$f(2,1,1,2)=1$	$f(1,1,1,0)=0$	$f(1,1,0,2)=0$
(24)	2	1	2	$f(2,1,2,2)=1$	$f(1,2,1,0)=1$	$f(2,1,1,2)=1$
(25)	**2**	**2**	**0**	$\boldsymbol{f(2,2,0,2)=0}$	$\boldsymbol{f(2,0,0,0)=0}$	$\boldsymbol{f(0,0,0,2)=1}$
(26)	2	2	1	$f(2,2,1,2)=2$	$f(2,1,2,0)=2$	$f(1,2,2,2)=0$
(27)	2	2	2	$f(2,2,2,2)=2$	$f(2,2,2,0)=0$	$f(2,2,0,2)=0.$

Получаем 27 вариантов, среди которых верным является 25-й вариант.

Теперь изменим атаку и улучшим результат. Введем следующий текст в устройство дешифрования:

$q_1q_1q_1q_1q_2q_2q_2q_2q_3q_3q_3q_3$ 000011112222

$q_2q_1q_1q_1q_3q_2q_2q_2q_1q_3q_3q_3$ 100021110222

$q_1q_2q_1q_1q_2q_3q_2q_2q_3q_1q_3q_3$ 010012112022

$q_1q_1q_2q_1q_2q_2q_3q_2q_3q_3q_1q_3$ или 001011212202

$q_1q_1q_3q_1q_2q_2q_1q_1q_3q_3q_2q_2$ 002011002211

$q_1q_2q_1q_3q_1q_3q_2q_1q_2q_3q_1q_2$ 010202101201

$q_3q_3q_2q_3q_2q_1q_3q_2q_3q_1q_1q_1$ 221210212000.

Таблица 4.7. Процесс дешифрования

$u_1 = {}^{(4,5)}f(l_1,l_2,l_3,v_1) = {}^{(4,5)}f(1,0,0,0) = 2$	$u_{43} = {}^{(4,5)}f(0,1,1,2) = 1$ **−(15)**
$u_2 = {}^{(4,5)}f(l_4,l_5,l_6,v_2) = {}^{(4,5)}f(2,1,1,0) = 1$	$u_{44} = {}^{(4,5)}f(1,1,2,1) = 1$ **−(44)**
$u_3 = {}^{(4,5)}f(l_7,l_8,l_9,v_3) = {}^{(4,5)}f(0,0,0,0) = 1$	$u_{45} = {}^{(4,5)}f(1,2,1,2) = 1$ **−(51)**
$u_4 = {}^{(4,5)}f(v_1,v_2,v_3,v_4) = {}^{(4,5)}f(0,0,0,0) = 1$ **−(1)**	$u_{46} = {}^{(4,5)}f(2,1,2,2) = 0$ **−(72)**
$u_5 = {}^{(4,5)}f(0,0,0,1) = 2$ **−(2)**	$u_{47} = {}^{(4,5)}f(1,2,2,0) = 2$ **−(52)**
$u_6 = {}^{(4,5)}f(0,0,1,1) = 0$ **−(5)**	$u_{48} = {}^{(4,5)}f(2,2,0,2) = 1$ **−(75)**
$u_7 = {}^{(4,5)}f(0,1,1,1) = 0$ **−(14)**	$u_{49} = {}^{(4,5)}f(2,0,2,0) = 1$ **−(61)**
$u_8 = {}^{(4,5)}f(1,1,1,1) = 1$ **−(41)**	$u_{50} = {}^{(4,5)}f(0,2,0,0) = 0$ **−(19)**
$u_9 = {}^{(4,5)}f(1,1,1,2) = 2$ **−(42)**	$u_{51} = {}^{(4,5)}f(2,0,0,2) = 2$ **−(57)**
$u_{10} = {}^{(4,5)}f(1,1,2,2) = 2$ **−(45)**	$u_{52} = {}^{(4,5)}f(0,0,2,0) = 2$ **−(7)**
$u_{11} = {}^{(4,5)}f(1,2,2,2) = 1$ **−(54)**	$u_{53} = {}^{(4,5)}f(0,2,0,1) = 1$ **−(20)**
$u_{12} = {}^{(4,5)}f(2,2,2,2) = 2$ **−(81)**	$u_{54} = {}^{(4,5)}f(2,0,1,1) = 2$ **−(59)**

$u_{13} = {}^{(4,5)}f(2,2,2,1) = 1$ **−(80)**	$u_{55} = {}^{(4,5)}f(0,1,1,0) = 2$ **−(13)**
$u_{14} = {}^{(4,5)}f(2,2,1,0) = 0$ **−(76)**	$u_{56} = {}^{(4,5)}f(1,1,0,0) = 2$ **−(37)**
$u_{15} = {}^{(4,5)}f(2,1,0,0) = 0$ **−(64)**	$u_{57} = {}^{(4,5)}f(1,0,0,2) = 1$ **−(30)**
$u_{16} = {}^{(4,5)}f(1,0,0,0) = 2$ **−(28)**	$u_{58} = {}^{(4,5)}f(0,0,2,2) = 1$ **−(9)**
$u_{17} = {}^{(4,5)}f(0,0,0,2) = 0$ **−(3)**	$u_{59} = {}^{(4,5)}f(0,2,2,1) = 2$ **−(26)**
$u_{18} = {}^{(4,5)}f(0,0,2,1) = 0$ **−(8)**	$u_{60} = {}^{(4,5)}f(2,2,1,1) = 1$ **−(77)**
$u_{19} = {}^{(4,5)}f(0,2,1,1) = 2$ **−(23)**	$u_{61} = {}^{(4,5)}f(2,1,1,0) = 1$ **−(67)**
$u_{20} = {}^{(4,5)}f(2,1,1,1) = 2$ **−(68)**	$u_{62} = {}^{(4,5)}f(1,1,0,1) = 2$ **−(38)**
$u_{21} = {}^{(4,5)}f(1,1,1,0) = 0$ **−(40)**	$u_{63} = {}^{(4,5)}f(1,0,1,0) = 0$ **−(31)**
$u_{22} = {}^{(4,5)}f(1,1,0,2) = 1$ **−(39)**	$u_{64} = {}^{(4,5)}f(0,1,0,2) = 0$ **−(12)**
$u_{23} = {}^{(4,5)}f(1,0,2,2) = 2$ **−(36)**	$u_{65} = {}^{(4,5)}f(1,0,2,0) = 0$ **−(34)**
$u_{24} = {}^{(4,5)}f(0,2,2,2) = 0$ **−(27)**	$u_{66} = {}^{(4,5)}f(0,2,0,2) = 2$ **−(21)**
$u_{25} = {}^{(4,5)}f(2,2,2,0) = 0$ **−(79)**	$u_{67} = {}^{(4,5)}f(2,0,2,1) = 2$ **−(62)**
$u_{26} = {}^{(4,5)}f(2,2,0,1) = 0$ **−(74)**	$u_{68} = {}^{(4,5)}f(0,2,1,0) = 1$ **−(22)**
$u_{27} = {}^{(4,5)}f(2,0,1,0) = 1$ **−(58)**	$u_{69} = {}^{(4,5)}f(2,1,0,1) = 1$ **−(65)**
$u_{28} = {}^{(4,5)}f(0,1,0,0) = 1$ **−(10)**	$u_{70} = {}^{(4,5)}f(1,0,1,2) = 2$ **−(33)**
$u_{29} = {}^{(4,5)}f(1,0,0,1) = 0$ **−(29)**	$u_{71} = {}^{(4,5)}f(0,1,2,0) = 2$ **−(16)**
$u_{30} = {}^{(4,5)}f(0,0,1,2) = 1$ **−(6)**	$u_{72} = {}^{(4,5)}f(1,2,0,1) = 2$ **−(47)**
$u_{31} = {}^{(4,5)}f(0,1,2,1) = 0$ **−(17)**	$u_{73} = {}^{(4,5)}f(2,0,1,2) = 0$ **−(60)**
$u_{32} = {}^{(4,5)}f(1,2,1,1) = 0$ **−(50)**	$u_{74} = {}^{(4,5)}f(0,1,2,2) = 1$ **−(18)**
$u_{33} = {}^{(4,5)}f(2,1,1,2) = 0$ **−(69)**	$u_{75} = {}^{(4,5)}f(1,2,2,1) = 0$ **−(53)**
$u_{34} = {}^{(4,5)}f(1,1,2,0) = 0$ **−(43)**	$u_{76} = {}^{(4,5)}f(2,2,1,2) = 2$ **−(78)**
$u_{35} = {}^{(4,5)}f(1,2,0,2) = 0$ **−(48)**	$u_{77} = {}^{(4,5)}f(2,1,2,1) = 2$ **−(71)**
$u_{36} = {}^{(4,5)}f(2,0,2,2) = 0$ **−(63)**	$u_{78} = {}^{(4,5)}f(1,2,1,0) = 2$ **−(49)**
$u_{37} = {}^{(4,5)}f(0,2,2,0) = 1$ **−(25)**	$u_{79} = {}^{(4,5)}f(2,1,0,2) = 2$ **−(66)**
$u_{38} = {}^{(4,5)}f(2,2,0,0) = 2$ **−(73)**	$u_{80} = {}^{(4,5)}f(1,0,2,1) = 1$ **−(35)**
$u_{39} = {}^{(4,5)}f(2,0,0,1) = 1$ **−(56)**	$u_{81} = {}^{(4,5)}f(0,2,1,2) = 0$ **−(24)**
$u_{40} = {}^{(4,5)}f(0,0,1,0) = 2$ **−(4)**	$u_{82} = {}^{(4,5)}f(2,1,2,0) = 1$ **−(70)**
$u_{41} = {}^{(4,5)}f(0,1,0,1) = 2$ **−(11)**	$u_{83} = {}^{(4,5)}f(1,2,0,0) = 1$ **−(46)**
$u_{42} = {}^{(4,5)}f(1,0,1,1) = 1$ **−(32)**	$u_{84} = {}^{(4,5)}f(2,0,0,0) = 0$ **−(55)**

На выходе получаем следующие 84 символа: 211120012212100200220120001101000001212211110211022122211211200022112220102222101 10.

Для полной реконструкции таблицы значений операции $^{(4,5)}f$ достаточно подать на вход 84 символа. Этот текст – лучший вариант с минимальной длиной для полного восстановления всех значений функции для случая 4-мерного группоида 3-го порядка.

Таким образом, для полной реконструкции значения таблицы операции $^{(i,n+1)}f$ и, следовательно, для таблицы операции f подтвержден результат, что для n -арного группоида требуемое количество символов составит $(\boldsymbol{n \cdot m^{n-1} + 1})(\boldsymbol{m - 1})$.

Минимальное количество символов в модифицированной атаке: $\boldsymbol{m^n + (n - 1)}$.

В результате проведения этих двух атак нам удается восстановить все значения функции дешифрования, но основная проблема заключается в подборе оптимальных текстов для группоидов разных степеней и порядков.

Определен еще один тип атаки, при котором минимальное количество символов составило: $\boldsymbol{m^{n-1} \cdot (m - 1)}$. Особенность этой атаки в том, что получены не все значения функции, а лишь достаточное количество для восстановления всей таблицы.

Что же касается атаки на выбранный шифротекст, то она может быть осуществлена только путем полного перебора всех значений функций, в которых фигурируют лидеры. Ситуация будет следующей: первые $(n - 1)$ символов, содержащие лидеров, могут принимать любые значения, а все остальные символы будут определяться по ним. Поэтому возможных вариантов дешифрируемых текстов будет m^n, где $n -$ арность, а $m -$ порядок i-обратимого группоида.

4.2. Атаки выбранным открытым текстом, построенным на основе i-обратимого n-арного группоида

Рассмотрим атаку открытым текстом, построенным с помощью n-арного группоида, обратимого на i-м месте, полученного с помощью обобщенного алгоритма Марковского.

Предположим, что у криптоаналитика есть доступ к шифровальному устройству, загруженному ключом. Он может построить следующий открытый текст ($n -$ арность и $m -$ это порядок i-обратимого группоида):

$$\underbrace{q_1 q_1 \dots q_1 q_1}_{n \text{ раз}} \underbrace{q_1 q_1 \dots q_1 q_2} \dots \underbrace{q_1 q_1 \dots q_1 q_m}$$

$$\underbrace{q_1 q_1 \dots q_2 q_1} \underbrace{q_1 q_1 \dots q_2 q_2} \dots \underbrace{q_1 q_1 \dots q_2 q_m}$$

$$\underbrace{q_1q_1\dots q_3q_1}\,\underbrace{q_1q_1\dots q_3q_2}\dots\underbrace{q_1q_1\dots q_3q_m}\dots$$

$$\underbrace{q_1q_1\dots q_mq_1}\,\underbrace{q_1q_1\dots q_mq_2}\dots\underbrace{q_1q_1\dots q_mq_m}\dots,$$

и ввести его в шифровальное устройство.

Количество символов, необходимое для восстановления таблицы шифрования, зависит от значений выбранных лидеров. Поэтому вопрос определения длины используемого открытого текста в каждом случае решается индивидуально.

Пример 4.2.1. Рассмотрим атаку открытым текстом для Примера 4.1.1. Выберем следующий открытый текст:

00000100201001101202002102 2

10010110211011111212012112 2

20020120221021121222022122 2 ...

Процесс шифрования текста и результаты имеют вид:

Таблица 4.8. Процесс шифрования

$v_1 = f(l_1, l_2, u_1) = f(1,2,0) = 1$	$v_{47} = f(2,2,2) = 2$ **−(27)**
$v_2 = f(l_3, l_4, u_2) = f(0,1,0) = 0$	$v_{48} = f(2,2,0) = 0$
$v_3 = f(v_1, v_2, u_3) = f(1,0,0) = 0$ **−(10)**	$v_{49} = f(2,0,1) = 0$
$v_4 = f(v_2, v_3, u_4) = f(0,0,0) = 0$ **−(1)**	$v_{50} = f(0,0,2) = 2$
$v_5 = f(0,0,0) = 0$	$v_{51} = f(0,2,1) = 2$
$v_6 = f(0,0,1) = 1$ **−(2)**	$v_{52} = f(2,2,1) = 1$
$v_7 = f(0,1,0) = 0$ **−(4)**	$v_{53} = f(2,1,2) = 1$
$v_8 = f(1,0,0) = 0$	$v_{54} = f(1,1,2) = 2$
$v_9 = f(0,0,2) = 2$ **−(3)**	$v_{55} = f(1,2,2) = 0$
$v_{10} = f(0,2,0) = 1$ **−(7)**	$v_{56} = f(2,0,0) = 2$
$v_{11} = f(2,1,1) = 0$ **−(23)**	$v_{57} = f(0,2,0) = 1$
$v_{12} = f(1,0,0) = 0$	$v_{58} = f(2,1,2) = 1$
$v_{13} = f(0,0,0) = 0$	$v_{59} = f(1,1,0) = 0$
$v_{14} = f(0,0,1) = 1$	$v_{60} = f(1,0,1) = 1$
$v_{15} = f(0,1,1) = 1$ **−(5)**	$v_{61} = f(0,1,2) = 2$
$v_{16} = f(1,1,0) = 0$ **−(13)**	$v_{62} = f(1,2,0) = 1$
$v_{17} = f(1,0,1) = 1$ **−(11)**	$v_{63} = f(2,1,2) = 1$
$v_{18} = f(0,1,2) = 2$ **−(6)**	$v_{64} = f(1,1,2) = 2$
$v_{19} = f(1,2,0) = 1$ **−(16)**	$v_{65} = f(1,2,1) = 2$
$v_{20} = f(2,1,2) = 1$ **−(24)**	$v_{66} = f(2,2,0) = 0$

$v_{21} = f(1,1,0) = 0$	$v_{67} = f(2,0,2) = 1$
$v_{22} = f(1,0,0) = 0$	$v_{68} = f(0,1,1) = 1$
$v_{23} = f(0,0,2) = 2$	$v_{69} = f(1,1,1) = 1$ **−(14)**
$v_{24} = f(0,2,1) = 2$ **−(8)**	$v_{70} = f(1,1,2) = 2$
$v_{25} = f(2,2,0) = 0$ **−(25)**	$v_{71} = f(1,2,1) = 2$
$v_{26} = f(2,0,2) = 1$ **−(21)**	$v_{72} = f(2,2,2) = 2$
$v_{27} = f(0,1,2) = 2$	$v_{73} = f(2,2,2) = 2$
$v_{28} = f(1,2,1) = 2$ **−(17)**	$v_{74} = f(2,2,2) = 2$
$v_{29} = f(2,2,0) = 0$	$v_{75} = f(2,2,0) = 0$
$v_{30} = f(2,0,0) = 2$ **−(19)**	$v_{76} = f(2,0,2) = 1$
$v_{31} = f(0,2,1) = 2$	$v_{77} = f(0,1,2) = 2$
$v_{32} = f(2,2,0) = 0$	$v_{78} = f(1,2,1) = 2$
$v_{33} = f(2,0,1) = 0$ **−(20)**	$v_{79} = f(2,2,2) = 2$
$v_{34} = f(0,0,1) = 1$	$v_{80} = f(2,2,2) = 2$
$v_{35} = f(0,1,0) = 0$	$v_{81} = f(2,2,2) = 2$
$v_{36} = f(1,0,2) = 2$ **−(12)**	$v_{82} = f(2,2,0) = 0$
$v_{37} = f(0,2,1) = 2$	$v_{83} = f(2,0,0) = 2$
$v_{38} = f(2,2,1) = 1$ **−(26)**	$v_{84} = f(0,2,0) = 1$
$v_{39} = f(2,1,0) = 2$ **−(22)**	$v_{85} = f(2,1,0) = 2$
$v_{40} = f(1,2,1) = 2$	$v_{86} = f(1,2,0) = 1$
$v_{41} = f(2,2,1) = 1$	$v_{87} = f(2,1,1) = 0$
$v_{42} = f(2,1,1) = 0$	$v_{88} = f(1,0,0) = 0$
$v_{43} = f(1,0,1) = 1$	$v_{89} = f(0,0,0) = 0$
$v_{44} = f(0,1,1) = 1$	$v_{90} = f(0,0,2) = 2 = v_9$
$v_{45} = f(1,1,2) = 2$ **−(15)**	$v_{91} = f(0,2,0) = 1 = v_{10}$
$v_{46} = f(1,2,1) = 2$	$v_{92} = f(2,1,1) = 0 = v_{11}$ …

На выходе шифровального устройства получаем следующие символы:

100001002100011012110022012202200102212210112220022112021101211220111

22222012222202121000210…

Таким образом, 26 символов из 27 будут восстановлены, а последний будет найден методом исключения. Пропустив даже все 81 символ таблица не будет восстановлена полностью. При этом начиная с 90-го символа начнется повтор (зацикливание). Одно из значений функции так и не определится.

Зная таблицу Кэли для операции f (Таблица 2.8, Пример 2.4.6), мы легко восстанавливаем таблицу для операции ${}^{(3,4)}f$ (Таблица 2.9, Пример 2.4.6).

Возьмем зашифрованный текст $202101 = v_1v_2v_3v_4v_5v_6$ и попытаемся взломать его:

$u_1 = {}^{(3,4)}f(l_1, l_2, v_1) = {}^{(3,4)}f(l_1, l_2, 2) \Longrightarrow u_1 = ?,$

$u_2 = {}^{(3,4)}f(l_3, l_4, v_2) = {}^{(3,4)}f(l_3, l_4, 0) \Longrightarrow u_2 = ?,$

$u_3 = {}^{(3,4)}f(v_1, v_2, v_3) = {}^{(3,4)}f(2,0,2) = 0 \Longrightarrow u_3 = 0,$

$u_4 = {}^{(3,4)}f(v_2, v_3, v_4) = {}^{(3,4)}f(0,2,1) = 0 \Longrightarrow u_4 = 0,$

$u_5 = {}^{(3,4)}f(v_3, v_4, v_5) = {}^{(3,4)}f(2,1,0) = 1 \Longrightarrow u_5 = 1,$

$u_6 = {}^{(3,4)}f(v_4, v_5, v_6) = {}^{(3,4)}f(1,0,1) = 1 \Longrightarrow u_6 = 1.$

Анализируя полученные результаты и используя таблицу значений функции ${}^{(3,4)}f$ получаем следующее: все элементы открытого текста однозначно идентифицируются, кроме первых двух (в которых присутствуют лидеры). Первые два элемента: ${}^{(3,4)}f(*,*,2)$ и ${}^{(3,4)}f(*,*,0)$ могут принимать любые значения.

Возможных значений открытого текста будет только $3^2 = 9$:

№	u_1	u_2	u_3	u_4	u_5	u_6
(1)	0	0	0	0	1	1
(2)	0	1	0	0	1	1
(3)	0	2	0	0	1	1
(4)	**1**	**0**	**0**	**0**	**1**	**1**
(5)	1	1	0	0	1	1
(6)	1	2	0	0	1	1
(7)	2	0	0	0	1	1
(8)	2	1	0	0	1	1
(9)	2	2	0	0	1	1

Среди них только 4-й вариант правильный. Первые два элемента не взломаны. Вопрос об определении лидеров в этом случае состоит в переборе $9^2 = 81$ вариантов.

Получаем для n-арного группоида в открытом тексте длины k, первые $(n-1)$ символов не взламываются, а остальные взламываются однозначно.

Выберем следующий открытый текст:

$q_1q_1q_1q_1q_2q_1q_2q_2q_1q_3q_1q_1$ 000010110200

$q_1q_3q_2q_3q_2q_1q_1q_3q_3q_3q_3q_2$ или 021210022221

$q_3q_2q_3q_2q_2$ 21211.

Процесс шифрования текста и результаты будут иметь вид:

Таблица 4.9. Процесс шифрования

$v_1 = f(l_1, l_2, u_1) = f(1,2,0) = 1$	$v_{16} = f(1,1,2) = 2$ **−(15)**
$v_2 = f(l_3, l_4, u_2) = f(0,1,0) = 0$	$v_{17} = f(1,2,1) = 2$ **−(17)**
$v_3 = f(v_1, v_2, u_3) = f(1,0,0) = 0$ **−(10)**	$v_{18} = f(2,2,0) = 0$ **−(25)**
$v_4 = f(v_2, v_3, u_4) = f(0,0,0) = 0$ **−(1)**	$v_{19} = f(2,0,0) = 2$ **−(19)**
$v_5 = f(0,0,1) = 1$ **−(2)**	$v_{20} = f(0,2,2) = 0$ **−(9)**
$v_6 = f(0,1,0) = 0$ **−(4)**	$v_{21} = f(2,0,2) = 1$ **−(21)**
$v_7 = f(1,0,1) = 1$ **−(11)**	$v_{22} = f(0,1,2) = 2$ **−(6)**
$v_8 = f(0,1,1) = 1$ **−(5)**	$v_{23} = f(1,2,2) = 0$ **−(18)**
$v_9 = f(1,1,0) = 0$ **−(13)**	$v_{24} = f(2,0,1) = 0$ **−(20)**
$v_{10} = f(1,0,2) = 2$ **−(12)**	$v_{25} = f(0,0,2) = 2$ **−(3)**
$v_{11} = f(0,2,0) = 1$ **−(7)**	$v_{26} = f(0,2,1) = 2$ **−(8)**
$v_{12} = f(2,1,0) = 2$ **−(22)**	$v_{27} = f(2,2,2) = 2$ **−(27)**
$v_{13} = f(1,2,0) = 1$ **−(16)**	$v_{28} = f(2,2,1) = 1$ **−(26)**
$v_{14} = f(2,1,2) = 1$ **−(24)**	$v_{29} = f(2,1,1) = 2$ **−(23)**
$v_{15} = f(1,1,1) = 1$ **−(14)**	

Результирующий текст в этом примере имеет минимальную длину. Эта атака использует 29 символов (найдены все 27 значений функций) вместо 69 символов (найдено только 26 из 27) в предыдущей версии.

Пример 4.2.2. Рассмотрим атаку открытым текстом для Примера 4.1.2 и выбираем следующий открытый текст:

0000 0001 0002 0010 0011 0012 0020 0021 0022
0100 0101 0102 0110 0111 0112 0120 0121 0122
0200 0201 0202 0210 0211 0212 0220 0221 0222
1000 1001 1002 1010 1011 1012 1020 1021 1022
1100 1101 1102 1110 1111 1112 1120 1121 1122
1200 1201 1202 1210 1211 1212 1220 1221 1222
2000 2001 2002 2010 2011 2012 2020 2021 2022
2100 2101 2102 2110 2111 2112 2120 2121 2122
2200 2201 2202 2210 2211 2212 2220 2221 2222
0000 0001 0002 001.

Процесс шифрования текста и результаты будут следующие:

Таблица 4.10. Процесс шифрования (фрагмент)

$v_4 = f(1,2,2,0) = 1 -$**(52)**	$v_{45} = f(1,1,2,0) = 0 -$**(43)**	$v_{119} = f(2,1,2,0) = 2 -$**(70)**
$v_5 = f(2,2,1,0) = 0 -$**(76)**	$v_{46} = f(1,2,0,1) = 0 - (\mathbf{47})$	$v_{121} = f(2,2,0,1) = 2 -$**(74)**
$v_6 = f(2,1,0,0) = 0 -$**(64)**	$v_{47} = f(2,0,0,0) = 0 -$**(55)**	$v_{128} = f(0,2,1,1) = 0 -$**(23)**
$v_7 = f(1,0,0,0) = 1 -$**(28)**	$v_{48} = f(0,0,0,2) = 1 - (\mathbf{3})$	$v_{131} = f(0,1,0,1) = 0 -$**(11)**
$v_8 = f(0,0,1,1) = 2 -$**(5)**	$v_{55} = f(1,2,1,1) = 2 - (\mathbf{50})$	$v_{132} = f(1,0,0,2) = 0 -$**(30)**
$v_9 = f(0,1,2,0) = 1 -$**(16)**	$v_{56} = f(2,1,2,1) = 0 - (\mathbf{71})$	$v_{133} = f(0,0,0,1) = 0 -$**(2)**
$v_{10} = f(1,2,1,0) = 1 -$**(49)**	$v_{57} = f(1,2,0,0) = 2 - (\mathbf{46})$	$v_{134} = f(0,0,0,0) = 2 -$**(1)**
$v_{11} = f(2,1,1,0) = 2 -$**(67)**	$v_{58} = f(2,0,2,1) = 0 - (\mathbf{62})$	$v_{135} = f(0,0,2,2) = 0 -$**(9)**
$v_{12} = f(1,1,2,2) = 2 -$**(45)**	$v_{62} = f(1,1,1,1) = 1 -$**(41)**	$v_{139} = f(0,1,1,2) = 0 - (\mathbf{15})$
$v_{15} = f(2,1,0,1) = 1 -$**(65)**	$v_{67} = f(0,2,0,2) = 2 - (\mathbf{21})$	$v_{143} = f(2,1,2,2) = 1 - (\mathbf{72})$
$v_{16} = f(1,0,1,0) = 0 -$**(31)**	$v_{71} = f(0,0,1,2) = 0 - (\mathbf{6})$	$v_{144} = f(1,2,1,2) = 0 -$**(51)**
$v_{17} = f(0,1,0,0) = 2 -$**(10)**	$v_{72} = f(0,1,0,2) = 1 -$**(12)**	$v_{161} = f(1,0,01) = 2 - (\mathbf{29})$
$v_{18} = f(1,0,2,0) = 0 -$**(34)**	$v_{79} = f(2,0,2,0) = 2 -$**(61)**	$v_{162} = f(0,0,2,1) = 2 -$**(8)**
$v_{19} = f(0,2,0,1) = 1 -$**(20)**	$v_{80} = f(0,2,2,1) = 0 - (\mathbf{26})$	$v_{173} = f(2,2,1,1) = 1 -$**(77)**
$v_{20} = f(2,0,1,1) = 0 -$**(59)**	$v_{86} = f(1,2,0,2) = 1 -$**(48)**	$v_{174} = f(2,1,1,1) = 0 -$**(68)**
$v_{24} = f(2,0,1,2) = 1 -$**(60)**	$v_{93} = f(2,0,1,0) = 2 - (\mathbf{58})$	$v_{179} = f(2,2,0,2) = 0 -$**(75)**
$v_{25} = f(0,1,1,0) = 1 -$**(13)**	$v_{94} = f(0,1,2,2) = 0 - (\mathbf{18})$	$v_{182} = f(0,2,2,2) = 1 -$**(27)**
$v_{26} = f(1,1,1,0) = 0 -$**(40)**	$v_{96} = f(2,0,0,2) = 2 -$**(57)**	$v_{192} = f(2,0,2,2) = 1 - (\mathbf{63})$
$v_{27} = f(1,1,0,2) = 0 -$**(39)**	$v_{97} = f(0,0,2,0) = 1 - (\mathbf{7})$	$v_{194} = f(2,1,0,2) = 2 -$**(66)**
$v_{29} = f(0,0,1,0) = 1 -$**(4)**	$v_{98} = f(0,2,1,2) = 1 - (\mathbf{24})$	$v_{212} = f(0,1,2,1) = 2 -$**(17)**
$v_{31} = f(1,1,1,2) = 2 -$**(42)**	$v_{99} = f(2,1,1,2) = 1 - (\mathbf{69})$	$v_{213} = f(1,2,2,1) = 2 -$**(53)**
$v_{32} = f(1,1,2,1) = 1 -$**(44)**	$v_{101} = f(1,1,0,0) = 1 -$**(37)**	$v_{214} = f(2,2,2,2) = 2 -$**(81)**
$v_{36} = f(1,2,2,2) = 0 -$**(54)**	$v_{102} = f(1,0,1,2) = 2 -$**(33)**	$v_{218} = f(2,2,2,0) = 0 -$**(79)**
$v_{37} = f(2,2,0,0) = 1 -$**(73)**	$v_{109} = f(1,0,1,1) = 1 - (\mathbf{32})$	$v_{288} = f(1,0,2,2) = 2 -$**(36)**
$v_{41} = f(0,2,0,0) = 0 -$**(19)**	$v_{116} = f(1,1,0,1) = 2 - (\mathbf{38})$	$v_{290} = f(2,2,1,2) = 2 -$**(78)**
$v_{42} = f(2,0,0,1) = 1 -$**(56)**	$v_{117} = f(1,0,2,1) = 1 -$**(35)**	$v_{338} = f(0,2,2,0) = 2 -$**(25)**
$v_{44} = f(0,1,1,1) = 2 - (\mathbf{14})$	$v_{118} = f(0,2,1,0) = 2 -$**(22)**	$v_{339} = f(2,2,2,1) = 1 -$**(80)**

На выходе шифровального устройства получаем следующие 339 символов:

122100121122101020102011100111211220

102001120001121121202011112020200101

010202201112010202012002111012000101

110111021220220102101000020011021210

111021200110210022020101122110022002
210020012021021201002011220101222222
201202200020220111112002002112222002
012101000020201200120010110122021022
122121100020210001220010120211112200
021221010202221.

Полную версию Таблицы 4.10 можно найти в Приложении 3, Таблица А3.18.

Для восстановления таблицы значений функции f (Приложение 3, Таблица А3.17) нам понадобилось 339 символов. Зная таблицу для операции f, легко восстанавливаем таблицу для операции ${}^{(4,5)}f$ (Приложение 3, Таблица А3.16)

Возьмем зашифрованный текст 202101001222 и попытаемся взломать его:

$u_1 = {}^{(4,5)}f(l_1, l_2, l_3, v_1) = {}^{(4,5)}f(l_1, l_2, l_3,\ 2) \Longrightarrow u_1 = ?,$

$u_2 = {}^{(4,5)}f(l_4, l_5, l_6, v_2) = {}^{(4,5)}f(l_4, l_5, l_6,\ 0) \Longrightarrow u_2 = ?,$

$u_3 = {}^{(4,5)}f(l_7, l_8, l_9, v_1) = {}^{(4,5)}f(l_7, l_8, l_9,\ 2) \Longrightarrow u_3 = ?,$

$u_4 = {}^{(4,5)}f(v_1, v_2, v_3, v_4) = {}^{(4,5)}f(2,0,2,1) = 2 \Longrightarrow u_4 = 2,$

$u_5 = {}^{(4,5)}f(v_2, v_3, v_4, v_5) = {}^{(4,5)}f(0,2,1,0) = 1 \Longrightarrow u_5 = 1,$

$u_6 = {}^{(4,5)}f(v_3, v_4, v_5, v_6) = {}^{(4,5)}f(2,1,0,1) = 1 \Longrightarrow u_6 = 1,$

$u_7 = {}^{(4,5)}f(v_4, v_5, v_6, v_7) = {}^{(4,5)}f(1,0,1,0) = 0 \Longrightarrow u_7 = 0,$

$u_8 = {}^{(4,5)}f(v_5, v_6, v_7, v_8) = {}^{(4,5)}f(0,1,0,0) = 1 \Longrightarrow u_8 = 1,$

$u_9 = {}^{(4,5)}f(v_6, v_7, v_8, v_9) = {}^{(4,5)}f(1,0,0,1) = 0 \Longrightarrow u_9 = 0,$

$u_{10} = {}^{(4,5)}f(v_7, v_8, v_9, v_{10}) = {}^{(4,5)}f(0,0,1,2) = 1 \Longrightarrow u_{10} = 1,$

$u_{11} = {}^{(4,5)}f(v_8, v_9, v_{10}, v_{11}) = {}^{(4,5)}f(0,1,2,2) = 1 \Longrightarrow u_{11} = 1,$

$u_{12} = {}^{(4,5)}f(v_9, v_{10}, v_{11}, v_{12}) = {}^{(4,5)}f(1,2,2,2) = 1 \Longrightarrow u_{12} = 1.$

Анализируя полученные результаты и используя таблицу значений функции ${}^{(4,5)}f$ получаем следующее: все элементы открытого текста однозначно идентифицируются, кроме первых трех (в которых фигурируют лидеры). Первые три элемента ${}^{(4,5)}f(*,*,*,2)$ и ${}^{(4,5)}f(*,*,*,0)$, могут принимать любые значения.

Возможных значений открытого текста будет $3^3 = 27$:

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
1)	0	0	0	2	1	1	0	1	0	1	1	1
2)	0	0	1	2	1	1	0	1	0	1	1	1

3)	0	0	2	2	1	1	0	1	0	1	1	1
4)	0	1	0	2	1	1	0	1	0	1	1	1
5)	0	1	1	2	1	1	0	1	0	1	1	1
6)	0	1	2	2	1	1	0	1	0	1	1	1
7)	0	2	0	2	1	1	0	1	0	1	1	1
8)	0	2	1	2	1	1	0	1	0	1	1	1
9)	0	2	2	2	1	1	0	1	0	1	1	1
10)	1	0	0	2	1	1	0	1	0	1	1	1
11)	1	0	1	2	1	1	0	1	0	1	1	1
12)	**1**	**0**	**2**	**2**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**
13)	1	1	0	2	1	1	0	1	0	1	1	1
14)	1	1	1	2	1	1	0	1	0	1	1	1
15)	1	1	2	2	1	1	0	1	0	1	1	1
16)	1	2	0	2	1	1	0	1	0	1	1	1
17)	1	2	1	2	1	1	0	1	0	1	1	1
18)	1	2	2	2	1	1	0	1	0	1	1	1
19)	2	0	0	2	1	1	0	1	0	1	1	1
20)	2	0	1	2	1	1	0	1	0	1	1	1
21)	2	0	2	2	1	1	0	1	0	1	1	1
22)	2	1	0	2	1	1	0	1	0	1	1	1
23)	2	1	1	2	1	1	0	1	0	1	1	1
24)	2	1	2	2	1	1	0	1	0	1	1	1
25)	2	2	0	2	1	1	0	1	0	1	1	1
26)	2	2	1	2	1	1	0	1	0	1	1	1
27)	2	2	2	2	1	1	0	1	0	1	1	1

Среди них верным является 12-й вариант. Вопрос об определении лидеров в этом случае состоит в переборе $27^3 = 5103$ вариантов.

Теперь выберем другой открытый текст:

0000 1111 2222 1100 0011

1120 2000 0010 0000 0022

1011 1111 2212 2220 2211

0000 1221 2000 1222 1222 2112.

Процесс шифрования текста и результаты будут иметь следующий вид:

Таблица 4.11. Процесс шифрования

$v_1 = f(l_1, l_2, l_3, u_1) = f(1,0,0,0) = 1$	$v_{43} = f(0,1,0,1) = 0 - (\mathbf{11})$
$v_2 = f(l_4, l_5, l_6, u_2) = f(2,1,1,0) = 2$	$v_{44} = f(1,0,0,1) = 2 - (\mathbf{29})$
$v_3 = f(l_7, l_8, l_9, u_3) = f(0,0,0,0) = 2$	$v_{45} = f(0,0,2,1) = 2 - \mathbf{(8)}$
$v_4 = f(v_1, v_2, v_3, u_4) = f(1,2,2,0) = 1 - \mathbf{(52)}$	$v_{46} = f(0,2,2,1) = 0 - (\mathbf{26})$
$v_5 = f(2,2,1,1) = 1 - \mathbf{(77)}$	$v_{47} = f(2,2,0,1) = 2 - \mathbf{(74)}$
$v_6 = f(2,1,1,1) = 0 - \mathbf{(68)}$	$v_{48} = f(2,0,2,1) = 0 - (\mathbf{62})$
$v_7 = f(1,1,0,1) = 2 - \mathbf{(38)}$	$v_{49} = f(0,2,0,2) = 2 - \mathbf{(21)}$
$v_8 = f(1,0,2,1) = 1 - \mathbf{(35)}$	$v_{50} = f(2,0,2,2) = 1 - (\mathbf{63})$
$v_9 = f(0,2,1,2) = 1 - \mathbf{(24)}$	$v_{51} = f(0,2,1,1) = 0 - (\mathbf{23})$
$v_{10} = f(2,1,1,2) = 1 - \mathbf{(69)}$	$v_{52} = f(2,1,0,2) = 2 - (\mathbf{66})$
$v_{11} = f(1,1,1,2) = 2 - \mathbf{(42)}$	$v_{53} = f(1,0,2,2) = 2 - (\mathbf{36})$
$v_{12} = f(1,1,2,2) = 2 - \mathbf{(45)}$	$v_{54} = f(0,2,2,2) = 1 - (\mathbf{27})$
$v_{13} = f(1,2,2,1) = 2 - \mathbf{(53)}$	$v_{55} = f(2,2,1,2) = 2 - (\mathbf{78})$
$v_{14} = f(2,2,2,1) = 1 - \mathbf{(80)}$	$v_{56} = f(2,1,2,0) = 2 - (\mathbf{70})$
$v_{15} = f(2,2,1,0) = 0 - \mathbf{(76)}$	$v_{57} = f(1,2,2,2) = 0 - (\mathbf{54})$
$v_{16} = f(2,1,0,0) = 0 - \mathbf{(64)}$	$v_{58} = f(2,2,0,2) = 0 - (\mathbf{75})$
$v_{17} = f(1,0,0,0) = 1 - \mathbf{(28)}$	$v_{59} = f(2,0,0,1) = 1 - (\mathbf{56})$
$v_{18} = f(0,0,1,0) = 1 - \mathbf{(4)}$	$v_{60} = f(0,0,1,1) = 2 - (\mathbf{5})$
$v_{19} = f(0,1,1,1) = 2 - \mathbf{(14)}$	$v_{61} = f(0,1,2,0) = 1 - (\mathbf{16})$
$v_{20} = f(1,1,2,1) = 1 - \mathbf{(44)}$	$v_{62} = f(1,2,1,0) = 1 - (\mathbf{49})$
$v_{21} = f(1,2,1,1) = 2 - \mathbf{(50)}$	$v_{63} = f(2,1,1,0) = 2 - \mathbf{(67)}$
$v_{22} = f(2,1,2,1) = 0 - \mathbf{(71)}$	$v_{64} = f(1,1,2,0) = 0 - \mathbf{(43)}$
$v_{23} = f(1,2,0,2) = 1 - \mathbf{(48)}$	$v_{65} = f(1,2,0,1) = 0 - \mathbf{(47)}$
$v_{24} = f(2,0,1,0) = 2 - \mathbf{(58)}$	$v_{66} = f(2,0,0,2) = 2 - \mathbf{(57)}$
$v_{25} = f(0,1,2,2) = 0 - \mathbf{(18)}$	$v_{67} = f(0,0,2,2) = 0 - \mathbf{(9)}$
$v_{26} = f(1,2,0,0) = 2 - \mathbf{(46)}$	$v_{68} = f(0,2,0,1) = 1 - \mathbf{(20)}$
$v_{27} = f(2,0,2,0) = 2 - \mathbf{(61)}$	$v_{69} = f(2,0,1,2) = 1 - \mathbf{(60)}$
$v_{28} = f(0,2,2,0) = 2 - \mathbf{(25)}$	$v_{70} = f(0,1,1,0) = 1 - \mathbf{(13)}$
$v_{29} = f(2,2,2,0) = 0 - \mathbf{(79)}$	$v_{71} = f(1,1,1,0) = 0 - (\mathbf{40})$
$v_{30} = f(2,2,0,0) = 1 - \mathbf{(73)}$	$v_{72} = f(1,1,0,0) = 1 - (\mathbf{37})$
$v_{31} = f(2,0,1,1) = 0 - \mathbf{(59)}$	$v_{73} = f(1,0,1,1) = 1 - (\mathbf{32})$
$v_{32} = f(0,1,0,0) = 2 - \mathbf{(10)}$	$v_{74} = f(0,1,1,2) = 0 - (\mathbf{15})$

$v_{33} = f(1,0,2,0) = 0 -$**(34)**	$v_{75} = f(1,1,0,2) = 0 - (\mathbf{39})$
$v_{34} = f(0,2,0,0) = 0 -$**(19)**	$v_{76} = f(1,0,0,2) = 0 - (\mathbf{30})$
$v_{35} = f(2,0,0,0) = 0 -$**(55)**	$v_{77} = f(0,0,0,1) = 0 - (\mathbf{2})$
$v_{36} = f(0,0,0,0) = 2 -$**(1)**	$v_{78} = f(0,0,0,2) = 1 - (\mathbf{3})$
$v_{37} = f(0,0,2,0) = 1 -$**(7)**	$v_{79} = f(0,0,1,2) = 0 - (\mathbf{6})$
$v_{38} = f(0,2,1,0) = 2 -$**(22)**	$v_{80} = f(0,1,0,2) = 1 - (\mathbf{12})$
$v_{39} = f(2,1,2,2) = 1 -$**(72)**	$v_{81} = f(1,0,1,2) = 2 - (\mathbf{33})$
$v_{40} = f(1,2,1,2) = 0 -$**(51)**	$v_{82} = f(0,1,2,1) = 2 - (\mathbf{17})$
$v_{41} = f(2,1,0,1) = 1 -$**(65)**	$v_{83} = f(1,2,2,1) = 2$
$v_{42} = f(1,0,1,0) = 0 -$**(31)**	$v_{84} = f(2,2,2,2) = 2 - (\mathbf{81})$

В этом примере мы используем 84 символа (находим 80 значений из 81) вместо 339 символов в предыдущей версии. В каждом случае выделение минимального текста – это достаточно сложная задача.

В заключение хотелось бы сказать несколько слов о криптоанализе шифров, построенных на основе Обобщенного Алгоритма 2. Этот криптоанализ представляет собой очень сложную задачу, в связи с тем, что отсутствует информация о степенях трансляций, используемых в алгоритме. Это еще раз указывает на то, что Обобщенный Алгоритм 2 более устойчив к криптоанализу, чем Обобщенный Алгоритм 1.

Учитывая проведенное исследование, можно сделать вывод, что оба обобщенных алгоритма представляют большой интерес для использования в криптографии.

4.3. Выводы по Главе 4

В четвертой главе описаны атаки с использованием выбранного шифротекста и выбранного открытого текста на шифрах, полученных с помощью обобщенного алгоритма Марковского (Обобщенного Алгоритма 1).

При проведении атак выбранным шифротекстом удалось определить количество используемых символов и в ряде случаев были получены тексты минимальной длины. Что касается атак с выбранным открытым текстом, здесь ситуация сложнее, поскольку каждый следующий символ зависит от предыдущего обработанного символа. Подбор текста минимальной длины в этом случае индивидуален для каждого примера. Сделан ряд важных оценок для всех проведенных атак.

На основании исследования, проведенного в Главе 4, и полученных результатов можно сделать следующие выводы:

1) Получено описание атак с использованием выбранного шифротекста и выбранного открытого текста на шифр, полученный с помощью обобщенного алгоритма Марковского (Обобщенного Алгоритма 1) [191, 192];
2) Для полной реконструкции таблицы значений операции ${}^{(i,n+1)}f$ (функции дешифрования) а следовательно, и таблицы значений операции f (функции шифрования) подтвержден результат, что для n-арного группоида, необходимое количество символов составило: $\boldsymbol{n \cdot m^{n-1}(m-1)+(m-2)}$, чтобы получить все значения соответствующей функции [192];
3) Минимальное количество символов в модифицированной атаке зашифрованным текстом будет: $\boldsymbol{m^n+n-1}$ символов, где $n-$ арность, а $m-$ порядок i-обратимого группоида [192];
4) Атака на шифротекст может быть осуществлена только путем полного перебора всех значений функций, в которых фигурируют лидеры. Общее количество этих значений равно $\boldsymbol{m^{n-1}}$;
5) Для n-арного группоида при взломе открытого текста длины k, первые $(n-1)$ символов не взламываются, а остальные определяются однозначно;
6) Значения используемых в алгоритме лидеров можно определить, только перебирая все возможные комбинации и это довольно сложная задача;
7) Для атаки выбранным открытым текстом удалось установить нижнее предельное значение необходимых символов для восстановления таблицы значений функции f. Остается открытым следующий вопрос: какой текст подавать на вход шифровального устройства, чтобы не превысить полученный лимит символов и всегда ли это будет возможно? [193]
8) Для группоидов третьего и четвертого порядка построены тексты наименьшей длины, но в каждом случае они подбирались индивидуально. Подбор такого текста– сложная задача [194, 195].

Анализируя результаты этой главы, можно сделать вывод, что криптоанализ шифров, построенных с использованием обобщенных алгоритмов Марковского, является достаточно трудоемкой задачей, требующей продолжения. В результате проведенного исследования возникли новые вопросы, представляющие интерес для криптоаналитиков.

В этой главе решаются задачи, связанные с криптоанализом шифров, построенных на основе i-обратимых n-арных группоидов во второй главе.

Результаты, представленные в Главе 4, были опубликованы в [191-195].

ОБЩИЕ ВЫВОДЫ И РЕКОМЕНДАЦИИ

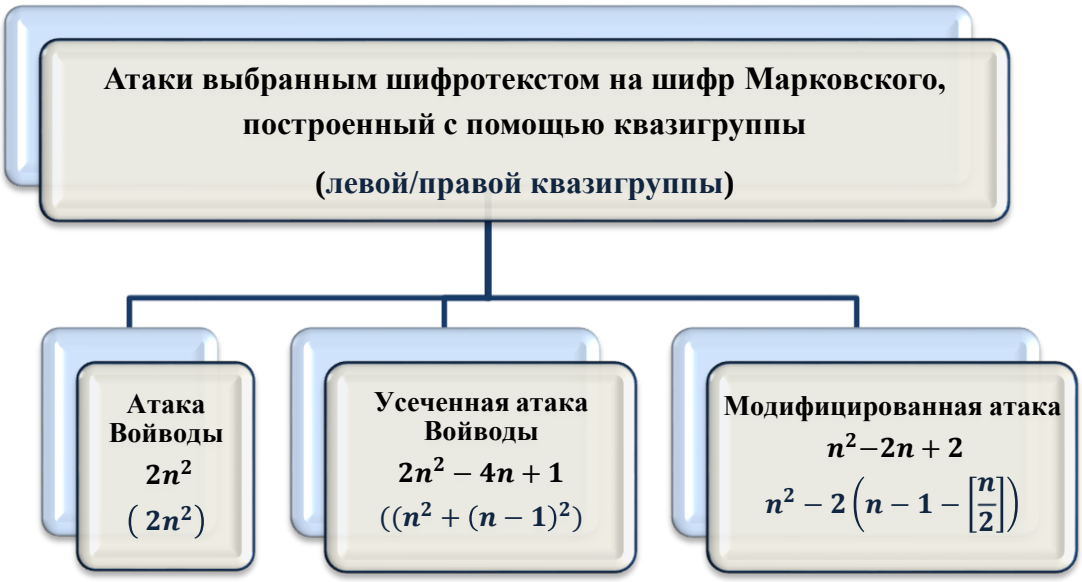
Исследование, выполненное в рамках докторской диссертации **«Использование информационных технологий в разработке криптографических и алгебраических алгоритмов»**, полностью соответствует цели и задачам, изложенным во вводной главе.

Результаты работы являются новыми и оригинальными. Разработаны и обобщены алгоритмы, позволившие улучшить работу классического алгоритма Марковского; изучены атаки на построенные шифры и показана степень стойкости этих шифров.

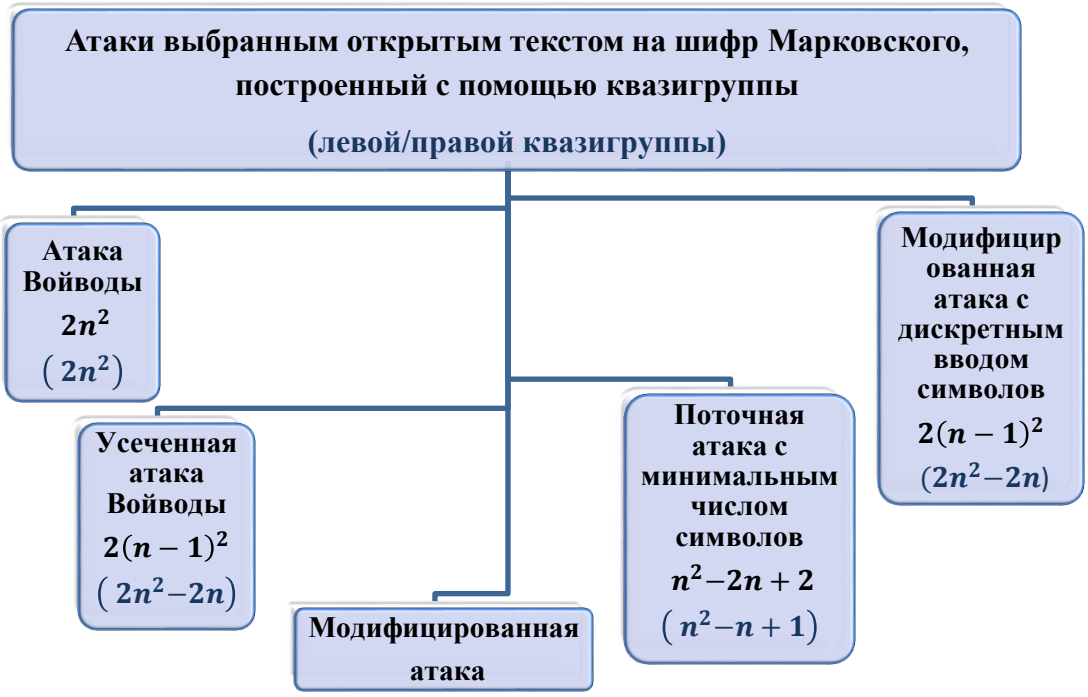
Теоретическая значимость диссертации определяется получением новых алгоритмов и шифров, построенных с использованием неассоциативных структур, таких как n-арные группоиды и квазигруппы. Прикладное значение диссертации заключается в использовании полученных результатов в теории кодирования и криптоанализе.

Анализируя полученные результаты, можно сделать следующие общие выводы:

1) Ключевой задачей защиты информации является создание надежных алгоритмов шифрования, поэтому любой вновь построенный алгоритм необходимо подвергать тщательному анализу с целью выявления его слабых мест и возможности взлома.
2) Использование квазигрупп в криптологии показывает лучшие возможности и результаты, чем использование ассоциативных систем.
3) Разработаны обобщенные алгоритмы Марковского для левой и правой квазигрупп и программы для их реализации (Приложение 2), которые имеют свои особенности и преимущества [171, 172].
4) Проведены атаки выбранным шифротекстом на шифры Марковского, построенные с использованием квазигрупп (левых и правых квазигрупп), проведен сравнительный анализ этих атак и предложены новые модифицированные атаки с улучшенными результатами. Отобраны тексты минимальной длины для каждой построенной атаки [187-190].

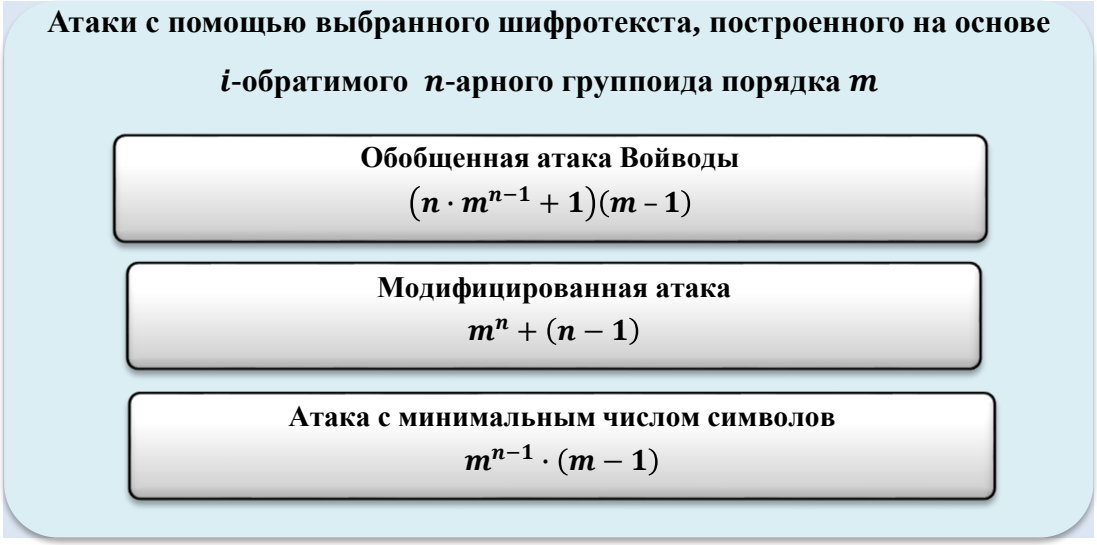


5) Проведены атаки выбранным открытым текстом на шифры Марковского, построенные с использованием квазигрупп, выявлены положительные и отрицательные стороны этих атак и предложены новые модифицированные атаки. Для потоковых атак выбранным открытым текстом определяется минимальное необходимое количество символов для полного восстановления таблицы квазигруппы шифрования (текст зависит от используемого лидера) [187-190].



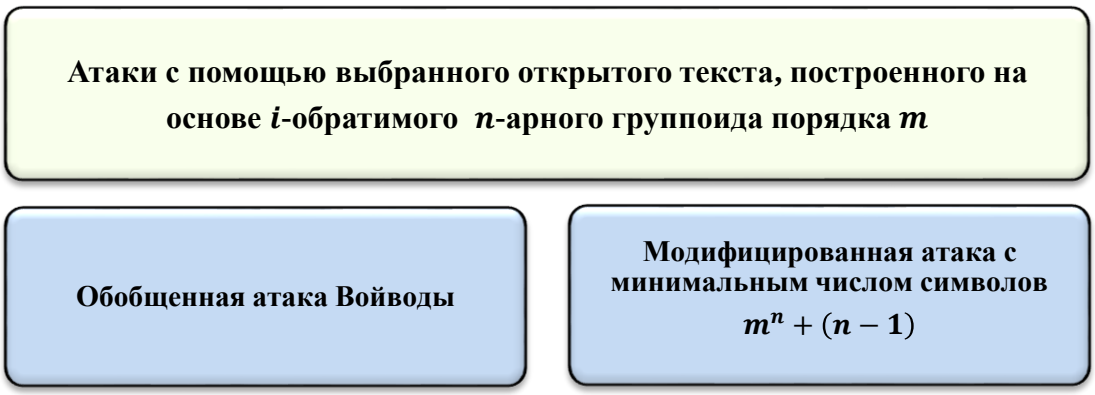
6) Построен обобщенный алгоритм Марковского для n -арного группоида, обратимого на одном фиксированном месте – Обобщенный Алгоритм 1 и разработаны программы, реализующие работу этого алгоритма [171, 173, 174].

7) Описаны атаки с использованием выбранного шифротекста на шифр, полученный с помощью обобщенного алгоритма Марковского (Обобщенного Алгоритма 1) [191, 192].



8) Атака на шифротекст может быть осуществлена путем полного перебора всех значений функций, в которых фигурируют лидеры. Общее количество этих значений равно m^{n-1}.

9) Описаны атаки с использованием выбранного открытого текста на шифр, полученный с помощью обобщенного алгоритма Марковского (Обобщенного Алгоритма 1) [191-193].



10) Для группоидов третьего и четвертого порядка построены тексты наименьшей длины, но в каждом случае они подбираются индивидуально. Подбор такого текста – сложная задача [194, 195].
11) Построен обобщенный алгоритм Марковского для n -арного группоида, обратимого на одном фиксированном месте, с использованием трансляций любых степеней – Обобщенный Алгоритм 2 и разработаны программы, реализующие работу этого алгоритма [174-176].
12) Общее количество необходимых лидеров для первого алгоритма будет равно: $(\boldsymbol{n-1})^{\boldsymbol{2}}$. Для второго алгоритма необходимое количество лидеров будет равно: $\frac{(\boldsymbol{n-1})\boldsymbol{n}}{\boldsymbol{2}}$. Второе число меньше первого на величину: $(\boldsymbol{n-1})\left(\frac{\boldsymbol{n}}{\boldsymbol{2}}-\boldsymbol{1}\right)$. Это говорит о преимуществе второго алгоритма перед первым (особенно с ростом числа n) [174].
13) Обобщенный Алгоритм 2 будет значительно сложнее, если мы помимо первой и второй степени трансляций будем использовать третью, четвертую и другие высшие степени. Особый интерес представляет определение обратных трансляций для тех, которые используются в Обобщенном Алгоритме 2 [174].
14) Проведен анализ всех разработанных программ, который включает в себя оценку наиболее важных параметров (среди них: длина текста, используемый алгоритм, количество необходимых лидеров, средняя скорость обработки данных, оценка сложности алгоритма с помощью концепции Big-O). В результате был сделан вывод, что программы работают успешно и имеют положительные характеристики.
15) Рассмотрен аналог системы шифрования Эль-Гамаля на основе алгоритма Марковского и изучены его особенности. Для него планируются новые модификации. Криптоанализ этой обобщенной схемы представляет собой сложную задачу, требующую решения [177-180].

Достоинства и ценность результатов диссертации. Предлагаемые разработки имеют значительную научную ценность благодаря высокой степени новизны и оригинальности. Полученные в работе результаты имеют теоретическую и прикладную ценность в таких областях, как алгебра, криптология и информатика.

Результаты автора, относящиеся к теме диссертации, опубликованы в [48, 104, 105, 111-113, 118, 171-180, 187-195].

Рекомендации. Полученные результаты могут быть использованы в различных областях и могут иметь практическое применение в теории кодирования.

На основании вышеизложенных выводов рекомендуется следующее:

- ✓ Особый интерес представляет продолжение применения алгоритма Марковского в теории кодирования и особенно в криптографии;
- ✓ Исследования по тематике диссертации могут быть продолжены как с алгебраической, так и с прикладной точек зрения. Особо важным является исследование возможностей использования квазигрупп и других неассоциативных систем в криптологии и теории кодирования;
- ✓ Построенные алгоритмы могут быть использованы в банковских информационных системах, а также при разработке различных банковских продуктов, как дополнительная защита, повышающая надежность и долговечность данных систем и продуктов (например, в современных пластиковых картах);
- ✓ Полученные результаты могут быть использованы для разработки алгебраических и криптографических алгоритмов в различных областях информатики;
- ✓ Содержание диссертации может служить основой для разработки спецкурсов для докторантов и магистрантов.

БИБЛИОГРАФИЯ

[1] FRIEDMAN, W. F. *The Index of Coincidence and Its Applications in Cryptanalysis*. Laguna Hills, California: Aegean Park Press, 1987. 101 p. ISBN 0-89412-137-5. Available: https://www.cryptomuseum.com/people/friedman/files/TIOC_Aegean_1987.pdf

[2] SHANNON, C. E. Communication theory of secrecy systems. In: *The Bell System Technical Journal*. 1949, vol. 28, no. 4, pp. 656-715. ISSN 0005-8580. Available: https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

[3] DIFFIE, W., HELLMAN M. New Directions in Cryptography. In: *IEEE Transactions on Information Theory*. 1976, vol.22, no.6, pp. 644-654. ISSN 0018-9448. Available: https://ee.stanford.edu/~hellman/publications/24.pdf

[4] БРАССАР, Ж. *Современная криптология*. Москва: ПОЛИМЕД, 1999. 178 с. ISBN 5-8832-010-2. Доступен: http://booksshare.net/books/physics/brassar-j/1999/files/sovremennayakritologiya1999.pdf

[5] MAGLIVERAS, STINSON, VAN TRUNG, T. New Approaches to Designing Public Key Cryptosystems Using One-Way Functions and Trapdoors in Finite Groups. In: *J. Cryptology*. 2002, vol.15, no.4, pp. 285-297. ISSN 1432-1378. Available: https://doi.org/10.1007/s00145-001-0018-3

[6] DEHORNOY, P. Braid-based cryptography. In: *Contemporary Mathematics, Group Theory, Statistics, and Cryptography*. 2004, vol. 360, pp. 5-33. ISBN 978-0-8218-3444-2. Available: https://www.lmno.cnrs.fr/archives/dehornoy/Surveys/Dgw.pdf

[7] DENES, J., KEEDWELL, A. D. Some applications of non-associative algebraic systems in cryptology. In: *Pure Mathematics and Applications*. 2001, vol. 12(2), pp.147-195. ISSN 1218-4586. Available: https://ideas.repec.org/a/cmt/pumath/puma2001v012pp0147-0195.html

[8] KOSCIELNY, Cz. *NLPN* Sequences over $GF(q)$. In: *Quasigroups and Related Systems*. 1997, vol.4, no.1, pp. 89-102. ISSN 1561-2848. Available: http://www.math.md/files/qrs/v4-n1/v4-n1-(pp89-102).pdf

[9] DENES, J., KEEDWELL, A. D. Latin Squares and their Applications. In: *Bulletin of the American mathematical society*. 1976, vol.82, no.3, pp. 468-471. ISSN 0273-0979. Available: https://www.ams.org/journals/bull/1976-82-03/S0002-9904-1976-14050-5/S0002-9904-1976-14050-5.pdf

[10] DENES, J., KEEDWELL, A.D. Latin squares: New Developments in the Theory and Applications. In: *Annals of Discrete mathematics*. 1991, North-Holland, vol. 46, pp. 1-469.

ISSN 0167-5060. ISBN 0 444 88899 3. Available: https://vdoc.pub/download/new-developments-in-the-theory-and-applications-2jbj5dahfhs0

[11] DENES, J. On Latin squares and a digital encrypting communication system. In: *P.U.M.A., Pure Mathematics and Applications*, Department of Mathematics, Corvinus University of Budapest. 2000, vol. 11, iss.4, pp.559-563. ISSN 1218-4586. Available: https://EconPapers.repec.org/RePEc:cmt:pumath:puma2000v011pp0559-0563

[12] KALKA, A. *Non-associative public-key cryptography*. 2012. 32 p. [Online]. Available: https://arxiv.org/pdf/1210.8270.pdf

[13] ТУЖИЛИН, М. Э. Латинские квадраты и их применение в криптографии. В: *Прикладная дискретная математика, Математические методы криптографии*. 2012, №3(17), с. 47-52. ISSN 2311-2263 (Online). Доступен: http://journals.tsu.ru/uploads/import/448/files/17-047.pdf

[14] МОВСИСЯН, Ю. Сверхтождества в алгебрах и многообразиях. В: *Успехи математических наук*. 1998, том 53, выпуск 1(319), с. 61-114. ISSN 0042-1316. Доступен: https://doi.org/10.4213/rm9

[15] ГРИБОВ, А.В., ЗОЛОТЫХ, П.А., МИХАЛЕВ, А.В. Построение алгебраической криптосистемы над квазигрупповым кольцом. В: *Математические вопросы криптографии*. 2010, том 1, выпуск 4, с. 23-32. ISSN 2220-2617. Доступен: https://doi.org/10.4213/mvk19

[16] MAZE, G., MONICO, C., ROSENTHAL, J. Public key cryptography based on semigroup actions. In: *Advances in Mathematics of Communications*. 2007, vol.1, no.4, pp.489-507. ISSN 1930-5346. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.3991&rep=rep1&type=pdf

[17] SHPILRAIN, V., USHAKOV, A. Thompson's Group and Public Key Cryptography. In: *Applied Cryptography and Network Security*, ACNS, Lecture Notes in Computer Science, Springer. 2005, vol. 3531, pp.151-163. ISBN 978-3-540-26223-7. ISSN 0302-9743. Available: https://doi.org/10.1007/11496137_11

[18] ATANI, R.E., ATANI, SH.E., MIRZAKUCHAKI, S. Public Key Cryptography Based on Semimodules over Quotient Semirings. In: *International Mathematical Forum*. 2007, vol.2, no.52, pp.2561-2570. ISSN 1312-7594. Available: http://www.m-hikari.com/imf-password2007/49-52-2007/ataniIMF49-52-2007.pdf

[19] KRAPEZ, A. Cryptographically Suitable Quasigroups via Functional Equations. In: *ICT Innovations 2012*, Advances in Intelligent Systems and Computing. 2013, vol. 207, pp. 265-274. ISSN 1857-7288. Available: https://doi.org/10.1007/978-3-642-37169-1_26

[20] KRAPEZ, A. ŠEŠELJA, B., TEPAVČEVIĆ, A. Solving linear equations by fuzzy quasigroups techniques. In: *Information Sciences*. 2019, vol. 491, pp.179-189. ISSN 0020-0255. Available: https://doi.org/10.1016/j.ins.2019.03.073

[21] MEYER, K. A. *A New Message Authentication Code Based on the Non-Associativity of Quasigroups*: PhD thesis of doctor of philosophy. Iowa State University, 2006. 91 p. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.146.3331&rep=rep1&type=pdf

[22] ARTAMONOV, V. Applications of quasigroups to cryprography. In: *Sarajevo Journal of Mathematics.* 2018, vol.14 (27), no.2, pp. 191–205. ISSN 1840-0655. Available: https://doi:10.5644/SJM. 14.2.07

[23] ARTAMONOV, V., CHAKRABARTI, S., MARKOV, V., PAL, S. Constructions of polynomially complete quasigroups of arbitrary order. In: *Journal of Algebra and Its Applications.* 2020, vol.20, no.12, 2150236. ISSN 0219-4988. Available: https://doi.org/10.1142/S0219498821502364

[24] KOSCIELNY, Cz., MULLEN, G.L. A quasigroup-based public-key cryptosystem. In: *International Journal of Applied Mathematics and Computer Science*. 1999, vol.9, no.4, pp. 955-963. ISSN 1641-876X.

[25] OCHODKOVA, E., SNASEL, V. Using quasigroups for secure encoding of file system. In: *Proceedings of the Conference Security and Protection of Information*, Abstract of Talks, Military Academy in Brno. 2001, pp. 175-181. ISBN 8085960281.

[26] MARKOVSKI, S., GLIGOROSKI, D., STOJCEVSKA, B. Secure two-way on-line communication by using quasigroup enciphering with almost public key. In: *Novi Sad Journal of Mathematics*. 2000, vol. 30, iss.2, pp. 43-49. ISSN 0352-0900. Available: https://eudml.org/doc/225166

[27] MARKOVSKI, S., GLIGOROSKI, D., BAKEVA, V. Quasigroup string processing: Part 1. In: *Proc. of Maked. Academ. of Sci. and Arts for Math. and Tech. Sci*. XX (1-2). 1999, pp.13-28. ISSN 1857-9027. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.5117&rep=rep1&type=pdf

[28] MARKOVSKI, S., DIMITROVA, V., TRAJCHESKA, Z., PETKOVSKA, M., KOSTADINOSKI, M., BUHOV, D. Block cipher defined by matrix presentation of

quasigroups. In: *IACR Cryptology ePrint Archive*. 2021, vol. 2021/ 1512, 3 p. Available: https://eprint.iacr.org/2021/1512

[29] SHCHERBACOV, V.A. *Elements of Quasigroup Theory and Applications*. 1st ed: Chapman and Hall/CRC, 2017. 598 p. ISBN 9781315120058. Available: https://doi.org/10.1201/9781315120058

[30] BAKEVA, V., DIMITROVA, V. Some probabilistic properties of quasigroup processed strings useful in cryptanalysis. In: *Communications in Computer and Information Science*. 2011, vol.83, pp. 61-70. ISSN 1865-0929. Available: https://link.springer.com/chapter/10.1007/978-3-642-19325-5_7

[31] BAKEVA, V., DIMITROVA, V., POPOVSKA-MITROVIKJ, A. Parastrophic quasigroup string processing. In: *Proceedings of the 8th Conference on Informatics and Information Technologies with International Participation*, 2011, Bitola, Macedonia, pp.19-21. Available: http://ciit.finki.ukim.mk/data/papers/8CiiT/8CiiT-05.pdf

[32] DIMITROVA, V., BAKEVA, V., POPOVSKA-MITROVIKJ, A., KRAPEZ, A. Classifications of quasigroups of order 4 by parastrophic quasigroups tranformation. In: *The International Mathematical Conference on Quasigroups and Loops, LOOPS'11*, Booklet of Abstracts, Trest, Czech Republic, 2011, p.6. Available: https://www2.karlin.mff.cuni.cz/~loops11/abstracts.pdf

[33] KRAPEZ, A., ZIVKOVIC, D. Parastrophically equivalent quasigroup equations. In: *Publications de l'Institut Mathématique*, Nouvelle Série, Beograd. 2010, vol.87(101), pp.39-58. ISSN 0350-1302. Available: https://www.emis.de/journals/PIMB/101/n101p039.pdf

[34] VOJVODA, M. *Stream Ciphers and Hash Functions: Analysis of Some New Design Approaches*: PhD thesis in technical sciences. Department of Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovak Republic, 2004. 94 p

[35] SHCHERBACOV, V.A., *Elements of quasigroup theory and some its applications in code theory and cryptology*, 2003. 85 p. [online]. Available: https://www2.karlin.mff.cuni.cz/~drapal/speccurs.pdf

[36] SHCHERBACOV, V.A. *On some known possible applications of quasigroups in cryptology*, 2003. 15 p. [online]. Available: https://www2.karlin.mff.cuni.cz/~drapal/krypto.pdf

[37] PETRESCU, A. Applications of quasigroups in cryptography. In: *Interdisciplinarity in Engineering Scientific International Conference Tg. Mures*-Romania, 15-16 November, 2007,

5 p. ISSN 2285-0945. Available: http://www.inter-eng.umfst.ro/2007/Papers/Section6/16-Petrescu-Quasigroups_pVI-16-1_5.pdf

[38] PETRESCU, A. *n* -Quasigroup cryptographic primitives: Stream ciphers. In: *Studia Universitatis Babeș-Bolyai Informatica*. 2010, vol. LV, iss.2, pp. 27-34. ISSN 1224-869X. Available: http://www.cs.ubbcluj.ro/~studia-i/contents/2010-2/03-Petrescu.pdf

[39] SHCHERBACOV, V.A. Quasigroups in cryptology. In: *Computer Science Journal of Moldova*. 2009, vol.17, no.2(50), pp. 193-228. ISSN 1561-4042. Available: http://www.math.md/files/csjm/v17-n2/v17-n2-(pp-193-228).pdf

[40] GLIGOROSKI, D., MARKOVSKI, S., KOCAREV, L. Edon-R, an infinite family of cryptographic hash functions. In: *International Journal of Network Security*. 2009, vol.8, no.3, pp. 293-300. ISSN 1816-353X. Available: http://ijns.jalaxy.com.tw/contents/ijns-v8-n3/ijns-2009-v8-n3-p293-300.pdf

[41] GLIGOROSKI, D., MARKOVSKI, S., KNAPSKOG, S. J. *A public key block cipher based on multivariate quadratic quasigroups*, 2008. 22 p. [Online]. https://arxiv.org/abs/0808.0247

[42] HASSINEN, M., MARKOVSKI, S. Secure SMS messaging using Quasigroup encryption and Java SMS API. In: *Proceedings of the Eighth Symposium on Programming Languages and Software Tools SPLST'03*, June 17-18, 2003, Kuopio, Finland, pp.187-200. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.2418&rep=rep1&type=pdf

[43] CHAKRABARTI, S, SAIBAL, K. P., SUGATA, G. An Improved 3-Quasigroup based Encryption Scheme. In: *ICT Innovations 2012, Secure and Intelligent Systems*, Web Proceedings, 2012, Ohrid, Macedonia, pp.173-184. ISSN 1857-7288. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.403.5151

[44] CSORGO, P., SHCHERBACOV, V. *On some quasigroup cryptographical primitives*, 2011. 11 p. [online]. https://arxiv.org/abs/1110.6591

[45] MOLDOVYAN, N.A., SHCHERBACOV, A.V., SHCHERBACOV, V.A. On some applications of quasigroups in cryptology. In: *Proceedings of the Workshop on Foundations of Informatics FOI-2015*, August 24-29, 2015, Chisinau, Republic of Moldova. pp.331-341. ISBN 978-9975-4237-3-1. Available: https://ibn.idsi.md/sites/default/files/imag_file/331_340_On%20some%20applications%20of%20quasigroups%20in%20cryptography.pdf

[46] ГРИБОВ, А.В. *Алгебраические неассоциативные структуры и их приложения в криптологии*: кандидатская диссертация, кандидата физико-математических наук,

МГУ, Москва, 2015. 93 с. Доступен: http://www.dslib.net/mat-logika/algebraicheskie-neassociativnye-struktury-i-ih-prilozhenija-v-kriptografii.html

[47] KOBLITZ, N. The uneasy relationship between mathematics and cryptography. In: *Notices of the American Mathematical Society*. 2007, vol.54, no.8, pp.972-979. ISSN 0002-9920. Available: https://www.ams.org/notices/200708/tx070800972p.pdf

[48] **МАЛЮТИНА, Н.,** ЩЕРБАКОВ, В. Роль математики в криптологии. В: *Материалы XI Международной научно-методической конференции «Совершенствование математического образования – 2020: состояние и перспективы развития»*, 2020, Тирасполь, с.15-19.

[49] УРБАНОВИЧ, П.П. *Защита информации методами криптографии, стеганографии и обфускации*. Минск: БГТУ, 2016. 220 с. ISBN 978-985-530-562-1. Доступен: https://elib.belstu.by/bitstream/123456789/23763/3/Urbanovich_zashhita.pdf

[50] SINGH, S. *The Code Book: The Secret History of Codes and Code-breaking*. Anchor; Reprint edition, 2011. 524 p. ISBN-13 978-0385495325.

[51] TOCCI, R., WIDMER, N., MOSS, G. *Digital Systems: Principles and Applications*. Pearson Prentice Hall, 2007. 970 p. ISBN 0-13-173969-7. Available: https://eceatglance.files.wordpress.com/2018/07/digital-systems-principles-and-applications-10th-edition-tocci-widmer.pdf

[52] GOLDWASSER, S., MIKALI, S. Probabilistic encryption. In: *Journal of Computer and System Sciences*. 1984, vol.28, no.2, pp.270-299. ISSN 0022-0000. Available: https://doi.org/10.1016/0022-0000(84)90070-9

[53] BLUM, M., GOLDWASSER, S. An Efficient Probabilistic Public Key Encryption Scheme which Hides All Partial Information. In: *Proceedings of Advances in Cryptology* - CRYPTO '84, Springer-Verlag, August, 1985, pp. 289-302. ISBN-10 0387156585. Available: https://dl.acm.org/doi/10.5555/19478.19501

[54] ALPERN, B., SCHNEIDER, F.B. Key Exchange Using “Keyless Cryptography”, In: *Information Processing Letters*. 1983, vol. 16, no.2, pp.79-81. ISSN 0020-0190. Available: https://hdl.handle.net/1813/6353

[55] YUNG, M. A secure and useful “keyless cryptosystem”. In: *Information Processing Letters*, 1985, vol. 21, no.1, pp.35-38. ISSN 0020-0190. Available: https://doi.org/10.1016/0020-0190(85)90106-1

[56] JOSEFSSON, S., LIUSVAARA, I. (2017). *Edwards-Curve Digital Signature Algorithm (EdDSA)*. Internet Engineering Task Force. ISSN 2070-1721. RFC 8032. Retrieved 2017-07-31. Available: https://www.rfc-editor.org/info/rfc8032

[57] BENNETT, C.H., BRASSARD, G. Quantum cryptography: Public-key distribution and coin tossing. In: *Theoretical Computer Science*, 2014, vol.560, no. 1, pp.7-11. ISSN 0304-3975. Available: https://doi.org/10.1016/j.tcs.2014.05.025

[58] BENNETT, C.H., BRASSARD, G. Quantum public key distribution reinvented, Association for Computing Machinery SIGACT News. 1987, vol. 18, no.4, pp. 51-53. ISSN 0163-5700. Available: https://doi.org/10.1145/36068.36070

[59] BENNETT, C.H., BESSETTE, F., BRASSARD, G., SALVAIL, L., SMOLIN, J. Experimental quantum cryptography. In: *Journal of Cryptology*. 1992, vol. 5. pp.3-28. ISSN 0933-2790. Available: https://link.springer.com/article/10.1007/BF00191318

[60] БАБЕНКО, Л.К., ИЩУКОВА, Е.А., МАРО, Е.А., СИДОРОВ, И.Д., КРАВЧЕНКО, П.П. Развитие криптографических методов и средств защиты информации. В: *Известия ЮФУ*. Технические науки. 2012, № 4 (129), с. 40-50. ISSN 1999-9429. Доступен: https://cyberleninka.ru/article/n/razvitie-kriptograficheskih-metodov-i-sredstv-zaschity-informatsii

[61] БАБЕНКО, Л.К., ИЩУКОВА, Е.А. *Современные алгоритмы блочного шифрования и методы их анализа*. Москва: Гелиос АРВ, 2006. 376 с. ISBN 5-85438-149-4. Доступен: https://sng1lib.org/ireader/2881021

[62] ПАНАСЕНКО, С.П. *Алгоритмы шифрования. Специальный справочник*. СПб.: БХВ-Петербург, 2009. 576 с. ISBN 978-5-9775-0319-8. Доступен: https://books.google.com.ua/books?id=Ha4AVrH9lSwC&printsec=frontcover&hl=ru#v=onepage&q&f=false

[63] АСОСКОВ, А.В. и др. *Потоковые шифры*. Книга 3-КУДИЦ-Образ, 2003. 336 с. ISBN 5-93378-078-2. Доступен: http://it-ebooks.ru/publ/it_secutity/potochnye_shifry/15-1-0-327

[64] Wikipedia contributors. Потоковый шифр [online]. Wikipedia The Free Encyclopedia. Последняя редакция: 14.02.2022, 11:02. [Цитируется 15.03.2022]. Доступен: https://ru.wikipedia.org/?curid=192553&oldid=120023352

[65] BIHAM, E., ANDERSON, R., KNUDSEN, L. Serpent: A new block cipher proposal. In: *Fast Software Encryption - FSE'98*, LNCS, Springer-Verlag Berlin Heidelberg, 1998, vol. 1372, pp. 222–238. ISBN 978-3-540-69710-7. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.8684&rep=rep1&type=pdf

[66] SHOR, P. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, 1996. 28 p. [Online]. Available: https://arxiv.org/abs/quant-ph/9508027

[67] Wikipedia contributors. PGP [online]. Wikipedia The Free Encyclopedia. Last edited: 16 May 2022, 12:05. [cited 02.06.2022]. Available: https://ru.wikipedia.org/?curid=24292&oldid=122340203

[68] БАБЕНКО, Л.К., ИЩУКОВА, Е.А., СИДОРОВ, И.Д. *Параллельные алгоритмы для решения задач защиты информации*. 2-е издание, стереотип. Москва: Горячая линия-Телеком, 2014. 304 с. ISBN 978-5-9912-0439-2. Доступен: https://sng1lib.org/book/2938454/5ba4b0

[69] БОГАТОВА, О.А., МАГОМЕДОВ, Ш.Г. Оценка эффективности решений в области криптографической безопасности для устройств интернета вещей. В: *Тенденции развития науки и образования*. 2019, №55(2), с.4-7. IDSP ljournal-10-2019-p6. SPLN 001-000001-0532-LJ. Доступен: https://doicode.ru/doifile/lj/55/lj-10-2019-20.pdf

[70] ШНАЙЕР, Б. *Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си*. Москва: Триумф, 2002. 816 с. ISBN 5-89392-055-4

[71] КАН, Д. *Взломщики кодов*. Москва: Центрполиграф, 2000. 480 с. ISBN 5-227-00678-4. Доступен: https://libking.ru/books/comp-/computers/125853-devid-kan-vzlomshchiki-kodov.html

[72] DAHL, O. M. *Exercise 1 question 1.4 Limitations and Differencies of using IPsec, TLS/SSL or SSH as VPN-solutio*. 2004. 5 p. [Online]. Available: https://www.semanticscholar.org/paper/Exercise-1-question-1-.-4-Limitations-and-of-using-Dahl/3505c68135246273403c9176c12752268ac0e262

[73] KNUDSEN, L.R. *Block Ciphers- Analysis, Design and Applications*: PhD dissertation. Aarhus University, 1994, DAIMI Report Series, 23(485). 269 p. ISSN 2245-9316. Available: https://doi.org/10.7146/dpb.v23i485.6978

[74] БЕРНИКОВ, В.О. Сравнительный анализ криптостойкости алгоритмов симметричного шифрования. В: *Труды БГТУ*, Сер. 3, Физико-математические науки и информатика, Минск, БГТУ. 2020, № 1 (230), с. 74-78. ISSN 2522-4638. Доступен: https://elib.belstu.by/handle/123456789/33420

[75] DIFFIE, W., HELLMAN, M.E. Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard. In: *Computer*. 1977, vol. 10, no. 6, pp. 74-84. ISSN 0018-9162. Available: https://ieeexplore.ieee.org/document/1646525

[76] DE CANNIÈRE, C., DUNKELMAN, O., KNEŽEVIĆ, M. KATAN and KTANTAN – A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: *Cryptographic Hardware and Embedded Systems - CHES 2009*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 2009, vol. 5747, pp. 272-288. Print ISBN 978-3-642-04137-2. Available: https://doi.org/10.1007/978-3-642-04138-9_20

[77] БАБЕНКО, Л.К., ИЩУКОВА, Е.А. Анализ симметричных криптосистем. В: *Известия Южного федерального университета*. Технические науки. 2012, №12 (137), с.136-147. ISSN 1999-9429. Доступен: https://cyberleninka.ru/article/n/analiz-simmetrichnyh-kriptosistem

[78] ДОЛГОВ, В.И., ОЛЕШКО О.И., ОЛЕЙНИКОВ Р.В. Дифференциальный криптоанализ. Сущность и проблемы использования. В: *Материалах Юбилейной научно-технической конференции*, Киев. 1998, с. 153-155. Доступен: https://science.donntu.edu.ua/ipz/golovko/library/dolgolesholey.pdf

[79] BIHAM, E., SHAMIR, A. Differential Cryptanalysis of the Full 16-round DES. In: *Advances in Cryptology - CRYPTO' 92*. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 1992, vol. 740, pp.487-496. Print ISBN 978-3-540-57340-1. Available: https://doi.org/10.1007/3-540-48071-4_34

[80] БАБЕНКО, Л.К., ИЩУКОВА, Е.А. Дифференциальный криптоанализ поточных шифров. В: *Известия Южного федерального университета*. Технические науки. 2009, №11(100), с. 232-238. ISSN 1999-9429. Доступен: https://cyberleninka.ru/article/n/differentsialnyy-kriptoanaliz-potochnyh-shifrov

[81] БАБЕНКО, Л.К., ИЩУКОВА, Е.А. Дифференциальный криптоанализ упрощенной функции хэширования SHA. В: *Известия Южного федерального университета*. Технические науки. 2010, №11(112), с. 203-220. ISSN 1999-9429. Доступен: https://cyberleninka.ru/article/n/differentsialnyy-kriptoanaliz-uproschennoy-funktsii-heshirovaniya-sha

[82] KNUDSEN, L.R., RIJMEN V., RIVEST, R.L., ROBSHAW, M.J.B. On the Design and Security of RC2. In: *Fast Software Encryption. FSE 1998*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 1998, vol. 1372, pp.206-221. Print ISBN 978-3-540-64265-7. Available: https://doi.org/10.1007/3-540-69710-1_14.

[83] BIRYUKOV, A., KUSHILEVITZ, E. Improved Cryptanalysis of RC5. In: *Advances in Cryptology — EUROCRYPT'98*, 1998. Lecture Notes in Computer Science, Springer, Berlin,

Heidelberg. 1998, vol. 1403, pp. 85–99. Print ISBN 978-3-540-64518-4. Available: https://doi.org/10.1007/BFb0054119

[84] SATOH, A., MORIOKA, S. Unified Hardware Architecture for 128-Bit Block Ciphers AES and Camellia. In: *Cryptographic Hardware and Embedded Systems - CHES 2003*. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 2003, vol. 2779, pp. 304-318. ISBN 978-3-540-40833-8. Available: https://doi.org/10.1007/978-3-540-45238-6_25

[85] BIHAM, E., SHAMIR, A. Differential cryptanalysis of DES-like cryptosystems. In: *Journal of Cryptology*. 1991, vol.4, no. 1, pp. 3-72. ISSN 0933-2790. Available: https://doi.org/10.1007/BF00630563

[86] HEYS, H. M. Linearly weak keys of RC5. In: *IEE Electronics Letters – IEEE*. 1997, vol. 33, iss. 10, pp. 836-838. ISSN 0013-5194. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.6473

[87] WU, W., FENG, D. Linear cryptanalysis of NUSH block cipher. In: *Science in China Series F: Information Sciences*, Science China Press, Springer-Verlag. 2008, vol. 45, iss. 1, pp.59-67. ISSN 1674-733X. Available: https://www.researchgate.net/publication/251372044_Linear_cryptanalysis_of_NUSH_block_cipher

[88] KNUDSEN, L. R., RADDUM, H. On Noekeon. In: *NESSIE New European Schemes for Signatures, Integrity, and Encryption*. Public report of the NESSIE project, Phase 1. 2001. 7 p. IST-1999-12324. Available: https://www.cosic.esat.kuleuven.be/nessie/reports/phase1/uibwp3-009.pdf

[89] ГРУШО, А.А., ПРИМЕНКО, Е.А., ТИМОНИНА, Е.Е. *Анализ и синтез криптоалгоритмов*. Курс лекций, Йошкар-Ола: Изд-во МФ МОСУ, 2000. 110 с. Доступен: https://www.docme.su/doc/1766645/grusho-a.a.-i-dr.---analiz-i-sintez-kriptoalgoritmov--2000

[90] COURTOIS, N., KLIMOV, A., PATARIN, J., SHAMIR, A. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In: *Advances in Cryptology - ASIACRYPT 2002*, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, 2002, Proceedings, vol. 2501 of Lecture Notes in Computer Science, pp. 267-287. ISBN 978-3-540-45539-4. Available: https://doi.org/10.1007/3-540-45539-6_27

[91] COURTOIS, N., PIEPRZYK, J. Cryptanalysis of block ciphers with Overdefined systems of equations. In: *Advances in Cryptology -ASIACRYPT*, 2002, Berlin, Germany, Springer,

Springer Nature, vol.2501, pp. 267-287. Print ISBN 3540001719. Available: https://eprint.iacr.org/2002/044.pdf

[92] COURTOIS, N. T., BARD, G.V. Algebraic Cryptanalysis of the Data Encryption Standard. In: *Cryptography and coding*:11th IMA International Conference, Cirencester, UK, 2007: proceedings, Berlin; New York: Springer, pp. 152-169. Print ISBN 978-3-540-77271-2. Available: https://doi.org/10.1007/978-3-540-77272-9_10

[93] KLEIMAN, E. *The XL and XSL attacks on Baby Rijndael*: PhD thesis in Computer Science. Iowa State University, 2005. 52 p. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.8626&rep=rep1&type=pdf

[94] БАБЕНКО, Л.К., МАРО, Е.А. Алгебраический криптоанализ упрощенного алгоритма шифрования Rijndael. В: *Известия Южного федерального университета*, Технические науки. 2009, № 11(100), с. 187-199. ISSN 1999-9429. Доступен: https://cyberleninka.ru/article/n/algebraicheskiy-kriptoanaliz-uproschennogo-algoritma-shifrovaniya-rijndael

[95] BIRYUKOV, A., WAGNER, D. Advanced Slide Attacks. In: *Advances in Cryptology — EUROCRYPT 2000*. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. 2000, vol. 1807, pp. 589-606. Print ISBN 978-3-540-67517-4. Available: https://doi.org/10.1007/3-540-45539-6_41

[96] Wikipedia contributors. История криптографии [online]. Wikipedia The Free Encyclopedia. Последняя редакция: 01.07.2022, 01:07. [Цитируется 09.07.2022]. Доступен: https://ru.wikipedia.org/?curid=2135710&oldid=123703540

[97] САБИНИН, Л.В. *Аналитические квазигруппы и геометрия*, ун-т Дружбы Народов. Москва: Букинист, 1991. 112 с. ISBN 5-209-00123-7

[98] LOHMUS, J., PAAL, E., SORGSEPP, L. About Nonassocoativity in Mathematics and Physics. In: *Acta Applicandae Mathematicae*. 1998, vol.50, iss. 1-2, pp. 3-31. ISSN 0167-8019. Available: https://doi.org/10.1023/A:1005854831381

[99] SABININ, L.V. Smooth quasigroups and loops. In: *Mathematics and its Applications*. Dordrecht, Kluwer Academic Publishers, 1999, vol. 492, 265 c. ISBN 978-0792359203

[100] ГВАРАМИЯ, А.А. Квазимногообразия автоматов. Связи с квазигруппами. В: *Сибирский математический журнал*. 1985, том 26, номер 3, с.11-30. ISSN 0037-4474. Доступен: http://www.mathnet.ru/links/c8a2d718712ea1ddb09d061abd94660d/smj7001.pdf

[101] ГОРЕЛИК, Г.Е. *Размерность пространства.* Москва: МГУ, 1983. 216 с.

[102] СУШКЕВИЧ, А. К. *Теория обобщенных групп*. Киев: ОНТИ НКТП, 1937. 176 с. Доступен: https://sng1lib.org/book/1018002/df1401

[103] BURSTIN, C., MAYER, W. Distributive Gruppen von endliher Ordnung. In: *Journal für die reine und angewandte Mathematik*. 1929, vol. 1929, iss. 160, pp.111-130. ISSN 1435-5345. Available: https://doi.org/10.1515/crll.1929.160.111

[104] CHERNOV, V., **MALYUTINA, N.,** SHCHERBACOV, V. Groupoids of order three with Bol-Moufang identities up to isomorphism. In: *Abstracts of International conference Mathematics & Information technologies: Research and Education, MITRE-2021*, July 1–3, 2021, Chisinau, Republic of Moldova, pp. 24-25. ISBN 978-9975-149-17-4. Available: https://ibn.idsi.md/sites/default/files/imag_file/24-25_33.pdf

[105] CHERNOV, V., DEMIDOVA, V., **MALYUTINA, N.,** SHCHERBACOV, V. Groupoids up to isomorphism of order three with some Bol-Moufang identities. In: *Proceedings of Workshop on Intelligent Information Systems WIIS2021*, October 14-15, 2021, Chisinau, Republic of Moldova, pp.85-88. ISBN 978-9975-68-438-5. Available: http://www.math.md/wiis2021/With_ISBN.pdf

[106] КУРОШ, А.Г. *Лекции по общей алгебре*. 2-е изд., Москва: Наука, 1973. 400 с. ISBN 978-5-8114-0617-3. Доступен: https://edu-lib.com/matematika-2/dlya-studentov/kurosh-a-g-lektsii-po-obshhey-algebre-onlayn

[107] БЕЛОУСОВ, В.Д. Уравновешенные тождества в квазигруппах. В: *Математический сборник*.1966, том 70(112), №1, с. 55-97. ISSN 0368-8666. Доступен: http://www.mathnet.ru/links/25da703243478a38774ae9d2e2724e3f/sm4214.pdf

[108] БЕЛЯВСКАЯ, Г.Б., ТАБАРОВ, А.Х. Характеристика линейных и алинейных квазигрупп. В: *Дискретная математика*. РАН, Москва. 1992, том 4, вып. 2., с. 142-147. ISSN 0234-0860. Доступен: http://mi.mathnet.ru/rus/dm/v4/i2/p142

[109] SHCHERBACOV, V.A. *On linear and inverse quasigroups and their applications in code theory*: thesis of doctor habilitat of physics and mathematics. Chisinau, 2008. 248 p.

[110] БЕЛОУСОВ, В.Д. О структуре дистрибутивных квазигрупп. В: *Математический сборник*. 1960, том 50(92), № 3, с. 267-298. ISSN 0368-8666. Доступен: http://mi.mathnet.ru/msb4793

[111] HOROSH, G., **MALYUTINA, N.,** SCERBACOVA, A., SHCHERBACOV, V. *Units in generalized derivatives of quasigroups*, 2020. 16 p. [online] Available: https://arxiv.org/pdf/2009.03605.pdf

[112] HOROSH, G., **MALYUTINA, N.,** SCERBACOVA, A., SHCHERBACOV, V. Units in generalized derivatives of quasigroups. In: *Communications of International Symposium "Actual Problems of Mathematics and Informatics" dedicated to the 90th birthday of professor Ion Valuţă*, November 27-28, 2020, Chişinău, Moldova, 2021. pp. 61-63. ISBN 978-9975-45-677-7. Available: https://ibn.idsi.md/sites/default/files/imag_file/61-63_23.pdf

[113] HOROSH, G., **MALYUTINA, N**., SCERBACOVA, A., SHCHERBACOV, V. Identities and generalized derivatives of quasigroups. In: *Computer Science Journal of Moldova.* 2022, vol. 30, no.2(89), pp. 170-186. ISSN 1561-4042. Available: https://www.math.md/files/csjm/v30-n2/v30-n2-(pp170-186).pdf

[114] BRANCIARD, C., GISIN, N., KRAUS, B., SCARANI, V. *Security of two quantum cryptography protocols using the same four qubit states*, 2005. 19 p. [Online]. Available: https://arxiv.org/abs/quant-ph/0505035

[115] PERLNER, R., COOPER, D. Quantum Resistant Public Key Cryptography: A Survey. In: *Proceedings of the 8th Symposium on Identity and Trust on the Internet*, April 2009, New York, NY: ACM, pp.85-93. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=901595

[116] ZBINDEN, H., GISIN, N., HUTTNER, B., MULLER, A., TITTEL, W. Practical aspects of quantum cryptographic key distributions. In: *Journal of Cryptology*. 2000, vol.13, iss.2, pp. 207-220. ISSN 0933-2790. Available: https://doi.org/10.1007/s001459910007

[117] BENNET, C. H., BRASSARD, G. Quantum cryptography: Public key distribution and coin tossing. In: *Theoretical Computer Science*. 2014, vol. 560, part 1, pp.7-11. ISSN 0304-3975. Available: https://doi.org/10.1016/j.tcs.2014.05.025

[118] **MALYUTINA, N.,** SHCHERBACOV, A., SHCHERBACOV, V. Construction of linear binary codes using orthogonal systems of Latin squares. In: *Proceedings of the Fifth Conference of Mathematical Society of Moldova, IMCS-55*, September 28- October 1, 2019, Chișinău, Republic of Moldova, pp. 97-100. ISBN 978-9975-68-378-4. Available: https://ibn.idsi.md/sites/default/files/imag_file/97-100_13.pdf

[119] SCHAUFFLER, R. *Eine Anwendung zyklischer Permutationen und ihre Theorie*: PhD thesis in mathematics. Philipps-Universitat zu Marburg, 1948. 114 p. (in German). Available: https://books.google.md/books/about/Eine_Anwendung_zyklischer_Permutationen.html?id=xO55nQEACAAJ&redir_esc=y

[120] DONOVAN, D. Critical sets in Latin squares of order less than 11. In: *JCMCC: The Journal of Combinatorial Mathematics and Combinatorial Computing*. 2002, vol. 29, pp.223-240.

ISSN 08353026. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.8703

[121] DONOVAN, D., HOWSE, A. Towards the spectrum of critical sets. In: *The Australasian Journal of Combinatorics*. 2000, vol.21, pp.107-130. ISSN 1034-4942. Available: https://ajc.maths.uq.edu.au/pdf/21/ocr-ajc-v21-p107.pdf

[122] KEEDWELL, D. Critical sets in latin squares: an intriguing problem. In: *The Mathematical Gazette*. 2001, vol. 85, no.503, pp. 239-244. ISSN 0025-5572. Available: https://www.jstor.org/stable/3622009

[123] KEEDWELL, D. Critical sets in latin squares and related matters: An update. In: *Utilitas Mathematica*. 2004, vol. 65, pp. 97-131. ISSN 03153681.

[124] MENEZES, A.J., VAN OORSCHOT, P.C., and VANSTONE, S.A. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, 1997. 810 p. ISBN 9780429466335. Available: https://doi.org/10.1201/9780429466335

[125] FEISTEL, H. Cryptography and Computer Privacy. In: *Scientific American*, Scientific American, a division of Nature America, Inc. 1973, vol. 228, no. 5, pp. 15-23. ISSN 00368733. Available: http://www.jstor.org/stable/24923044

[126] LAI X., MASSEY J.L. A Proposal for a New Block Encryption Standard. In: *Advances in Cryptology - EUROCRYPT '90*. Lecture Notes in Computer Science, 1990, vol. 473, Springer, Berlin, Heidelberg, pp. 389-404. Print ISBN 978-3-540-53587-4. Available: https://doi.org/10.1007/3-540-46877-3_35

[127] KOSCIELNY, C. A method of constructing quasigroup-based stream-ciphers. In: *The International Journal of Applied Mathematics and Computer Science*. 1996, vol.6, no.1, pp.109-121. ISSN 1641-876X. Available: https://www.amcs.uz.zgora.pl/?action=paper&paper=1150

[128] KOSCIELNY, C. Generating quasigroups for cryptographic applications. In: *The International Journal of Applied Mathematics and Computer Science*. 2002, vol.12, no.4, pp.559-569. ISSN 1641-876X. Available: https://www.amcs.uz.zgora.pl/?action=paper&paper=117

[129] COPPERSMITH, D., Weakness in quaternion signatures. In: *Journal of Cryptology*. 2001, vol.14, iss.2, pp.77-85. ISSN 0933-2790. Available: https://doi.org/10.1007/s001450010006

[130] DENES, J., KEEDWELL, A. D., A new authentication scheme based on latin squares. In: *Discrete Mathematics*. 1992, vol.106-107, pp.157-161. ISSN 0012-365X. Available: https://doi.org/10.1016/0012-365X(92)90543-O

[131] DAWSON, Ed., DONOWAN, D., OFFER, A. Ouasigroups, Isotopisms and Authentification Schemes. In: *The Australasian Journal of Combinatorics*. 1996, vol.13, pp.75-88. ISSN 1034-4942. Available: https://ajc.maths.uq.edu.au/pdf/13/ocr-ajc-v13-p75.pdf

[132] SARVATE, D.G., SEBERRY, J. Encryption methods based on combinatorial designs. In: *Ars Combinatoria*, 21-A. 1986, pp. 237-245. ISSN 0381-7032. Available: https://ro.uow.edu.au/infopapers/1019/

[133] DENES, J., DENES, T. Non-associative algebraic system in cryptology. Protection against "meet in the middle" attack. In: *Quasigroups and Related Systems*. 2001, vol. 8, no.1, pp.7-14. ISSN 1561-2848. Available: https://www.math.uci.edu/~brusso/denescrhptography.pdf

[134] DRAPAL, A. Hamming distances of groups and quasi-groups. In: *Discrete Mathematics*. 2001, vol.235, iss.1-3, pp.189-197. ISSN 0012-365X. Available: https://doi.org/10.1016/S0012-365X(00)00272-7

[135] DRAPAL, A. On Groups that Diff er in One of Four Squares. In: *European Journal of Combinatorics*. 2002, vol.23, iss.8, pp.899-918. ISSN 0195-6698. Available: https://doi.org/10.1006/eujc.2002.0615

[136] DRAPAL, A., ZHUKAVETS, N. On multiplication tables of groups that agree on half of the columns and half of the rows. In: *Glasgow Mathematical Journal*. 2003, vol. 45, iss.2, pp.293-308. ISSN 0017-0895. Available: https://doi:10.1017/S0017089503001253

[137] ШАПОШНИКОВ, И. Г. О конгруэнциях конечных многоосновных универсальных алгебр. В: *Дискретная математика*. 1999, том 11, выпуск 3, с.48-62. ISSN 0234-0860. Доступен: https://doi.org/10.4213/dm387

[138] КАРГАПОЛОВ, М.И. и МЕРЗЛЯКОВ, Ю.И. *Основы теории групп*. 3-е изд, Москва: Наука, 1982. 288 с. Доступен: http://www.vixri.ru/d/Kargapolov%20M.I.,Merzljakov%20Ju.I.__Osnovy%20teorii%20Grupp%20+%20KOUROVSKAJa%20TETRAD.pdf

[139] БЕЛОУСОВ, В.Д. *n-арные квазигруппы*. Кишинев: Штиинца, 1972. 225 с. Доступен: https://sng1lib.org/book/2982147/7d8570

[140] MARINI, A., SHCHERBACOV, V.A. On autotopies and automorphisms of n-ary linear quasigroups. In: *Journal Algebra and Discrete Mathematics*. 2004, no.2, pp. 59-83. ISSN 1726-3255. Available: https://www.researchgate.net/publication/228565543_On_autotopies_and_automorphisms_of_n-ary_linear_quasigroups

[141] NOSOV, V.A., PANKRATIEV, A.E. Latin squares over Abelian groups. In: *Journal of Mathematical Sciences*. 2008, vol.149, iss.3, pp. 1230-1234. ISSN 10723374. Available: https://doi.org/10.1007/s10958-008-0061-9

[142] MARKOVSKI, S., MILEVA, A. Generating huge quasigroups from small non-linear bijections via extended Feistel function. In: *Quasigroups and Related Systems*. 2009, vol. 17, no.1, pp.91-106. ISSN 1561-2848. Available: http://www.quasigroups.eu/

[143] COOPER, J., DONOVAN, D., SEBERRY, J. Secret sharing schemes arising from latin squares. In: *Bulletin of the Institute of Combinatorics and its Applications*. 1994, vol.12, pp. 33-43. ISSN 1183-1278. Available: https://www.academia.edu/20806021/Secret_sharing_schemes_arising_from_Latin_squares

[144] DONOVAN, D.M., LEFEVRE, J.G., McCOURT, T.A., CAVENAGH, N.J., KHODKAR, A. Identifying flaws in the security of critical sets in latin squares via triangulations. In: *Australasian Journal of Combinatorics*. 2012, vol.52, pp.243–268. ISSN 1034-4942. Available: https://espace.library.uq.edu.au/view/UQ:316953

[145] FALCON, R.M. Latin squares associated to principal autotopisms of long cycles. Application in Cryptography. In: *Proceedings of Transgressive Computing 2006*: a conference in honor of Jean Della Dora, 24-26 April 2006, pp.213-230. Available: https://idus.us.es/handle/11441/69179

[146] FALCON, R.M. *Cycle structures of autotopisms of the Latin squares of order up to 11*, 2009. 18 p. [online] Available: https://arxiv.org/pdf/0709.2973v2.pdf.

[147] STONES, D.S., VOJTECHOVSKY, P., WANLESS, I.M. Cycle structure of autotopisms of quasigroups and Latin squares. In: *Journal of Combinatorial Designs*. 2012, vol.20, no.5, pp.227-263. ISSN 1520-6610. Available: https://users.monash.edu.au/~iwanless/papers/auttJCD.pdf

[148] BELYAVSKAYA, G.B. Secret-sharing schemes and orthogonal systems of k -ary operations. In: *Quasigroups and Related Systems*. 2009, vol. 17, no.2(22), pp.161-176. ISSN 1561-2848. Available: http://www.math.md/files/qrs/v17-n2/v17-n2-(pp161-176).pdf

[149] GOLOMB, S.W., WELCH, L.R., and DENES, J. *Encryption system based on crossed inverse quasigroups*, 2007. United States patent, No. US 7,280,663 B1. Available: https://patents.google.com/patent/US7280663B1/en

[150] KEEDWELL, A.D. Crossed-inverse quasigroups with long inverse cycles and applications to cryptography. In*: The Australasian Journal of Combinatorics*. 1999, vol.20, pp.241-250. ISSN 1034-4942. Available: https://ajc.maths.uq.edu.au/pdf/20/ocr-ajc-v20-p241.pdf

[151] KEEDWELL, A.D., SHCHERBACOV, V.A. On m-inverse loops and quasigroups with a long inverse cycle. In: *The Australasian Journal of Combinatorics*. 2002, vol.26, pp.99-119. ISSN 1034-4942. Available: https://ajc.maths.uq.edu.au/pdf/26/ajc_v26_p099.pdf

[152] KEEDWELL, A.D., SHCHERBACOV, V.A. Construction and properties of (r,s,t)-inverse quasigroups, In: *Discrete Mathematics*. 2003, vol.266, iss.1-3, pp.275-291. ISSN 0012-365X. Available: https://doi.org/10.1016/S0012-365X(02)00814-2

[153] KEEDWELL, A.D., SHCHERBACOV, V.A. Construction and properties of (r,s,t)-inverse quasigroups, II. , In: *Discrete Mathematics*. 2004, vol.288, iss.1-3, pp.61-71. ISSN 0012-365X. Available: https://doi.org/10.1016/j.disc.2004.06.020

[154] KEEDWELL, A.D., SHCHERBACOV, V.A. Quasigroups with an inverse property and generalized parastrophic identities. In: *Quasigroups and Related Systems*. 2005, vol. 13, no.1, pp.109-124. ISSN 1561-2848. Available: http://www.math.md/files/qrs/v13-n1/v13-n1-(pp109-124).pdf

[155] SHCHERBACOV, V.A. On definitions of groupoids closely connected with quasigroups. In: *Buletinul Academiei de Stiinte a Republicii Moldova*, Matematica. 2007, Number 2(54), pp. 43-54. ISSN 1024-7696. Available: http://www.math.md/files/basm/y2007-n2/y2007-n2-(pp43-54).pdf

[156] MOLDOVYAN, N.A., MOLDOVYANU, P.A. New primitives for digital signature algorithms. In: *Quasigroups and Related Systems*. 2009, vol.17, no.2, pp.271-282. ISSN 1561-2848. Available: http://www.quasigroups.eu/contents/download/2009/17_20.pdf

[157] RIVEST, R.L. All-or-nothing encryption and the package transform. In: *Fast Software Encryption FSE 1997*, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 1997, vol. 1267, pp. 210-218. ISSN 0302-9743. Available: https://doi.org/10.1007/BFb0052348

[158] MARNAS, S.I., ANGELIS, L., BLERIS, G.L. All-or-nothing transforms using quasigroups. In: *Proceedings of 1st Balkan Conference in Informatics (BCI)*, Thessaloniki, Greece, 2003, pp.183-191. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.9502&rep=rep1&type=pdf

[159] ELGAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. In: *IEEE Transactions on Information Theory*. 1985, vol.31, no.4, pp. 469-472. ISSN 0018-9448. Available: https://doi.org/10.1109/TIT.1985.1057074

[160] Wikipedia contributors. ElGamal encryption [online]. Wikipedia The Free Encyclopedia. Last edited: 13 April 2022, 09:16. [cited 20.05.2022]. Available: https://en.wikipedia.org/w/index.php?title=ElGamal_encryption&oldid=1082465022

[161] SHCHERBACOV, V.A. On the structure of left and right F-, SM- and E-quasigroups. In: *Journal of Generalized Lie Theory and Applications*. 2009, vol. 3, no.3, pp.197-259. ISSN 1736-5279. Available: https://arxiv.org/pdf/0811.1725.pdf

[162] MAZE, G. *Algebraic Methods For Constructing One-Way Trapdoor Functions:* PhD thesis of doctor of philosophy. University of Notre Dame, 2003. 151 p. Available: http://user.math.uzh.ch/maze/Articles/DissJoli.pdf

[163] SHOR, P.W. Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer. In: *SIAM Journal on Computing*. 1997, vol.26, no.5, pp.1484-1509. ISSN 0097-5397. Available: http://mmrc.amss.cas.cn/tlb/201702/W020170224608150589788.pdf

[164] GENTRY, C. *A Fully Homomorphic Encryption Scheme*: PhD thesis of doctor of philosophy. The Department of Computer Science of Stanford University, 2009. 209 p. Available: https://crypto.stanford.edu/craig/craig-thesis.pdf

[165] GENTRY, C. Computing arbitrary functions of encrypted data. In: *Communications of the ACM*. 2010, vol.53, no.3, pp. 97-105. ISSN 0001-0782. Available: https://crypto.stanford.edu/craig/easy-fhe.pdf

[166] SHCHERBACOV, V.A. *Quasigroup based crypto-algorithms*, 2012. 23 p. [online]. Available: https://arxiv.org/pdf/1201.3016.pdf

[167] MARKOVSKI, S., GLIGOROSKI, D., ANDOVA, S. Using quasigroups for one-one secure encoding. In: *Proceedings of the VIII international conference on logic and computer science: theoretical foundations of computer science-LIRA'97*,1997, Novi Sad, pp.157-167. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.6590&rep=rep1&type=pdf

[168] РЯБКО, Б.Я., ФИОНОВ, А.Н., *Криптографические методы защиты информации*: учебное пособие. Москва: Горячая линия-Телеком, 2005. 229 с. ISBN 5-89176-233-1. Доступен: http://cyber.sibsutis.ru/%D0%A1%D0%9F%D0%98/%D0%9F%D0%B5%D1%80%D0%B2%D0%B0%D1%8F%20%D1%87%D0%B0%D1%81%D1%82%D1%8C/%D0%9A%D0%9C%D0%97%D0%98.pdf

[169] КАТЫШЕВ, С.Ю., МАРКОВ, В.Т., НЕЧАЕВ, А.А. Использование неассоциативных группоидов для реализации процедуры открытого распределения ключей. В: *Дискретная математика*.2014, том 26, выпуск 3, с.45-64. ISSN 0234-0860. Доступен: https://doi.org/10.4213/dm1289

[170] STICKEL, E. A new method for exchanging secret keys. In: *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, 2005, vol.

2, pp. 426-430. Print ISBN 0-7695-2316-1. Available: https://ieeexplore.ieee.org/document/1488999

[171] MALYUTINA, N. The role of quasigroups in cryptosystems. A generalisation of Markovski algorithm. In: *Proceedings of International Conference "Contemporary Trends in Science Development: Visions of Young Researchers"*, Edition 7, Vol.1, June 15, 2018, Chisinau, Republic of Moldova: Tipogr. "Biotehdesign", 2018, pp. 20-26. ISBN 978-9975-108-45-4. Available: https://ibn.idsi.md/sites/default/files/imag_file/20-26_0.pdf

[172] МАЛЮТИНА, Н.Н., ЩЕРБАКОВ, В.А. Роль квазигрупп в криптосистемах. Алгоритм Марковского. В: *Вестник Приднестровского университета* Сер.: Физико-математические и технические науки. 2018, Вып.3 (60), с. 53-57. ISSN 1857-1174. Доступен: http://spsu.ru/images/files/science/vestnik/Vestnik_Fis-mat_3-18.pdf

[173] МАЛЮТИНА, Н.Н., ЩЕРБАКОВ, В.А. Роль квазигрупп в криптосистемах. Обобщение алгоритма Марковского. In: *Communications of International Conference on Mathematics, Informatics and Information Technologies dedicated to the Illustrious Scientist Valentin Belousov, MITI2018*, April 19-21, 2018, Balti, Republic of Moldova, pp. 88-89. ISBN 978-9975-3214-7-1. Доступен: http://lib.udau.edu.ua/bitstream/123456789/6677/1/MITI2018_117.pdf

[174] МАЛЮТИНА, Н.Н., ЩЕРБАКОВ, В.А. Обобщения алгоритма Марковского. В: *Вестник Приднестровского университета* Сер.: Физико-математические и технические науки. 2019, Вып.3 (63), с. 55-64. E-ISSN 2345-1548. Доступен: http://spsu.ru/images/files/science/vestnik/Vestnik_Fis-mat_3-18.pdf

[175] MALYUTINA, N., SCERBACOVA A., SHCHERBACOV, V. *Markovski algorithm on i-invertibile groupoid*s, 2018. 3 p. [onlain] Available: https://arxiv.org/pdf/1806.02267.pdf

[176] MALYUTINA, N., SCHERBACOVA A., SHCHERBACOV, V. Some generalisations of Markovski algorithm on i-invertibile groupoids. In: *Proceedings of the Conference on Mathematical Foundations of Informatics, MFOI 2018*, July 2-6, 2018, Chisinau, Republic of Moldova, pp. 149-153. ISBN 978–9975–4237–7–9. Available: https://ibn.idsi.md/sites/default/files/imag_file/149-153_1.pdf

[177] МАЛЮТИНА, Н. Об аналоге схемы Эль-Гамаля на основе квазигрупп. В: *The materials of the student scientific conference with international participation*. Edition LXX-a, Vol.2, April 28, 2021, Chisinau, Tiraspol State University, pp.269-275. ISBN 978-9975-76-337-0. Доступен: https://ibn.idsi.md/sites/default/files/imag_file/2p-269-275.pdf

[178] MALYUTINA, N., SHCHERBACOV, V. *An analogue of the ElGamal scheme based on the Markovski algorithm*, 2021. 10 p. [online] Available: https://arxiv.org/pdf/2111.08476.pdf

[179] MALYUTINA, N., SHCHERBACOV V. An analogue of the ElGamal scheme based on the Markovski algorithm. In: *ROMAI Journal*. 2021, vol.17, no.1, pp. 105-114. ISSN 2065-7714. Available: https://rj.romai.ro/arhiva/2021/1/Malytutina-Shcherbacov.pdf

[180] MALYUTINA, N. An analogue of the El Gamal scheme based on the Markovski algorithm. In: *Contemporary Research and Evaluation Methodologies, Biological and Chemical Sciences Physical and Mathematical Sciences Economic Sciences*. April 22-23, 2021, Chisinau: CEP USM, pp. 112-118. ISBN: 978-9975-159-16-6. Available: https://ibn.idsi.md/sites/default/files/imag_file/p-112-118_0.pdf

[181] GROSEK, O. On the Stability of Stream Ciphers (O stabilite prúdových šifier). In: *Proceedings of the conference "Jesenný seminár z kryptoanalýzy"*, Lipt. Mikuláš. 1996, pp.14-26. (in Slovak).

[182] NEMOGA, K. Linear recurring sequences (Lineárne rekurentné postupnosti). In: *Proceedings of the conference "Jesenný seminár z kryptoanalýzy"*, Lipt. Mikuláš. 1996, pp.1-13 (in Slovak).

[183] GROSEK, O., NEMOGA, SATKO, L. Ideal difference tables from an algebraic point of view, Cryptology and Information Security. In: *Proceedings of VI RECSI*, Teneriffe, Spain, 2000, RAMA, Madrid, pp. 453-454.

[184] VOJVODA, M. Cryptanalysis of a file encoding system based on quasigroup. In: *Journal of Electrical Engineering*. 2003, vol.54, no.12. ISSN 1335-3632. Available: https://www.researchgate.net/publication/268889584_Cryptanalysis_of_a_file_encoding_system_based_on_a_quasigroup

[185] VOJVODA, M. *Attacks on a file encryption system based on quasigroup*. In: *Proceedings of Elitech 2003*, Department of Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology,2003, Bratislava, Slovak Republic, pp. 54-56.

[186] HORÁK, P., ALDRED, R.E.L., FLEISCHNER, H. Completing Latin Squares: Critical Sets, In: *Journal of Combinatorial Designs*. 2002, vol.10, iss.6, pp.419-432. ISSN 1063-8539. Available: https://ur.booksc.eu/book/1500768/f8c3dd

[187] MALYUTINA, N. Cryptanalysis of some stream ciphers. In: *Abstracts of International conference "Mathematics & Information technologies: Research and Education" (MITRE-*

2019), June 24-26, 2019, Moldova State University, Chișinău, Republic of Moldova, pp. 45-46. ISBN 978-9975-149-17-4. Available: http://cecmi.usm.md/mitre/en/past-conferences

[188] MALYUTINA, N. Cryptanalysis of some stream ciphers. In: *Proceedings of International Conference "Contemporary trends in the development of science: visions of young researchers"*, Edition 8, Vol.1, June 10, 2019, Chisinau, Republic of Moldova: Biotehdesign, pp. 9-14. ISBN 978-9975-108-66-9. Available: https://ibn.idsi.md/sites/default/files/imag_file/9-14_4.pdf

[189] MALYUTINA, N. Cryptanalysis of some stream ciphers. In: *Quasigroups and Related Systems*. 2019, vol.27, no.2, pp. 281-292. ISSN 1561-2848. Available: http://www.quasigroups.eu/contents/download/2019/27_26.pdf

[190] МАЛЮТИНА, Н.Н., ЩЕРБАКОВ, В.А. Криптоанализ некоторых потоковых шифров. В: *Вестник Приднестровского университета* Сер.: Физико-математические и технические науки. 2020, Вып.3 (66), с. 37-48. E-ISSN 2345-1548. Доступен: http://spsu.ru/images/files/science/vestnik/%D0%92%D0%B5%D1%81%D1%82%D0%BD%D0%B8%D0%BA_3_2020_1.pdf

[191] MALYUTINA, N. Attacks on Generalized Markovski Crypto-algorithm. In: *Proceedings of the 5th Conference on Mathematical Foundations of Informatics*, MFOI 2019, July 3-6, 2019, Iași, Romania, pp. 87-103. ISBN 978-606-714-481-9. Available: http://mfoi2019.info.uaic.ro/PDF/MFOI2019_Volume.pdf

[192] MALYUTINA, N. Cryptanalysis of some stream ciphers based on n-ary groupoids. In: *Quasigroups and Related Systems*. 2020, vol.28, no.2, pp. 251-268. ISSN 1561-2848. Available: http://www.quasigroups.eu/contents/download/2020/28_21.pdf

[193] МАЛЮТИНА, Н.Н., ЩЕРБАКОВ, В.А. Криптоанализ некоторых потоковых шифров, построенных на основе n-арных группоидов, обратимых на i-м месте. В: *Вестник Приднестровского университета* Сер.: Физико-математические и технические науки. 2021, Вып.3 (69), с. 62-74. E-ISSN 2345-1548. Доступен: http://spsu.ru/images/files/science/3-2021.pdf

[194] MALYUTINA, N. Cryptanalysis of some stream ciphers based on n-ary groupoids. In: *Proceedings of International Conference "Contemporary Trends in Science Development: Visions of Young Researchers.* Edition IX, Vol.1, June 10, 2020, Chisinau, Republic of Moldova: Biotehdesign, pp. 35-40. ISBN 978-9975-108-66-9. Available: https://ibn.idsi.md/sites/default/files/imag_file/35-40_32.pdf

[195] **MALYUTINA, N.** Cryptanalysis of some stream ciphers. In: *LOOPS 2019, Conference Budapest University of Technology and Economics,* July 7- July 13, 2019, Hungary, pp.34-37. Available: https://algebra.math.bme.hu/LOOPS19/l19_booklet_final.pdf

[196] АХМЕТОВ, Б. А. и др. *Прикладная криптология: методы шифрования*: Учебное пособие. Алматы: КазНИТУ им.К.И. Сатпаева, 2015. 496 с. ISBN 978-601-228-879-7. Доступен: https://er.nau.edu.ua/handle/NAU/32584

[197] MERKLE, R., HELLMAN, M. Hiding Information and Signatures in Trapdoor Knapsacks, In: *IEEE Transactions on Information Theory.* 1978, vol. 24, no. 5, pp. 525-530. ISSN 0018-9448. Available: https://ieeexplore.ieee.org/document/1055927

[198] ЖУКОВ, А.Е. Легковесная криптография, Часть 2, В: *Вопросы кибербезопасности*. 2015, №2(10), с. 2-10. ISSN 2311-3456. Доступен: https://cyberrus.com/wp-content/uploads/2015/05/vkb_10_01.pdf

[199] THE 128-BIT BLOCKCIPHER CLEFIA. Algorithm Specification, January 2010. 64 p. Available: https://www.cryptrec.go.jp/en/cryptrec_13_spec_cypherlist_files/PDF/22_00espec.pdf

[200] RIVEST, R. The MD4 Message-Digest Algorithm, RFC 1320, IETF tools, April 1992. Available: https://www.rfc-editor.org/info/rfc1320

[201] SCHNEIER, B. *Applied Cryptography Second Edition: Protocols, Algorithms, and Source Code in C (cloth)*, John Wiley & Sons, Inc, 2nd edition, 1996. 1027 p. ISBN 0471128457. Available: https://mrajacse.files.wordpress.com/2012/01/applied-cryptography-2nd-ed-b-schneier.pdf

ПРИЛОЖЕНИЯ

Приложение 1. Характеристики некоторых криптографических алгоритмов

Алгоритм DES использует множественные арифметико-логические преобразования исходного текста. DEA – это блочный алгоритм, работающий с 64-битными блоками данных. В нем используется ключ длиной 56 бит и, кроме них, вычисляются 8 битов четности. Расшифровка в DES осуществляется в обратном порядке. Важнейшим параметром алгоритма является не только ключ, но и преобразование блока открытого текста или шифротекста с помощью S-матриц. У алгоритма есть проблема слабых ключей, поскольку существуют пары разных ключей, которые при шифровании преобразуют открытый текст в один и тот же зашифрованный текст. В настоящее время алгоритм находит ограниченное применение. Одним из очевидных преимуществ алгоритма DES является относительно высокая скорость его реализации из-за небольшого размера ключа. Одним из направлений совершенствования алгоритма является использование базового DES в качестве компонента другого алгоритма. Дополнительную информацию можно найти в [49, 62, 196].

Алгоритм Люцифера. Этот алгоритм представляет собой последовательность перестановок и подстановок. Основные блоки аналогичны алгоритму DES. S -блоки алгоритма Люцифера имеют 4-битные входы и выходы. Бит ключа используется для выбора одного из двух возможных блоков. Этот алгоритм значительно менее устойчив, чем DES [196].

Алгоритм IDEA. Этот алгоритм работает с 64-битными блоками данных и 128-битными ключами. Этот алгоритм является подстановочно-перестановочным алгоритмом. Нововведением в алгоритме является использование операций из разных алгебраических групп. Размер ключа IDEA больше, чем у DES, но меньше, чем у Blowfish. Скорость шифрования IDEA на Intel486SX/33MHz в 2 раза выше, чем у DES, но почти в 2 раза ниже, чем у Blowfish [196].

Алгоритм Blowfish разработан Б. Шнайером. Алгоритм предназначен для линий связи, где не предусмотрена частая смена ключей. Он работает с 64-битными блоками данных. Ключ может быть расширен до 448 бит. Алгоритм обычно состоит из 16 раундов, в каждом из которых выполняются зависящие от ключа операции замены и перестановки. Формально этот алгоритм является сетью Фейстеля. Более подробно этот алгоритм описан в [62].

Алгоритм распределения ключей Диффи-Хеллмана. Абоненты могут использовать этот алгоритм для обмена ключевой информацией по открытым каналам. Протокол Диффи-Хеллмана уязвим для атаки «человек посередине»: злоумышленник может перехватить открытое сообщение и отправить вместо него собственное. Таким образом, он получит общие секретные ключи и сможет читать и изменять все передаваемые сообщения [62].

Алгоритм Knapsack, разработанный Р. Мерклем и М. Хеллманом, был первым алгоритмом шифрования с открытым ключом общего назначения [197]. Алгоритм основан на идее решения ряда задач упаковки ранца. Существуют две различные задачи упаковки ранца: одна из них легко решается и характеризуется линейным ростом трудоемкости, а другая, сложнее. Простой в хранении ранец можно превратить в сложный. В качестве открытого ключа можно использовать сложный в упаковке ранец, который легко использовать для шифрования, но невозможно для расшифровки сообщений. А в качестве закрытого ключа мы можем использовать легко разгадываемый ранец, который обеспечивает простой способ расшифровки сообщения. Тому, кто не знает закрытый ключ, придется попытаться решить непростую задачу по упаковке ранца.

RSA – это первый полноценный алгоритм с открытым ключом, который можно использовать как для шифрования, так и для создания цифровых подписей, названный в честь трех изобретателей – Ривеста, Шамира и Адльмана. Этот алгоритм уже много лет противостоит криптоаналитическим атакам и является самым популярным. Безопасность алгоритма основана на трудоемкости факторизации больших чисел. Открытый и закрытый ключи являются функциями двух больших простых чисел, состоящих из 100-200 десятичных цифр и более. Стойкость шифрования напрямую связана с размером ключа, а удвоение размера ключа обеспечивает экспоненциальное увеличение криптографической стойкости, хотя и снижает производительность. Ключи RSA обычно имеют длину 1024 или 2048 бит. Следует отметить, что в 2013 году группа криптографов под руководством Шамира успешно определила 4096-битный ключ RSA с помощью акустического криптоанализа, но любой алгоритм шифрования уязвим для этого метода атаки [198]. Аппаратный RSA более чем в 1000 раз медленнее, чем DES. При программной реализации обоих алгоритмов производительность первого из них примерно в 100 раз хуже.

Криптография на эллиптических кривых является альтернативой RSA при реализации шифрования с открытым ключом и позволяет обеспечить более высокий уровень безопасности при меньшей вычислительной мощности, что делает ее более подходящей для мобильных приложений. Эта система была предложена в 1985 году Нилом Коблицем и

Виктором Миллером. Защищенность системы основана на сложности решения задачи дискретного логарифмирования в группе точек эллиптической кривой над конечным полем, чем и обусловлена ее высокая криптостойкость. Эллиптические кривые – это математический объект, который можно определить над любым полем. Еще одним преимуществом криптосистем на эллиптических кривых является высокая скорость обработки информации. Алгоритм ECC позволяет обеспечить сравнимый уровень безопасности при использовании ключа меньшего размера [199]. Уровень безопасности, который достигается в RSA при использовании 1024-битных ключей, в системах на эллиптических кривых реализуется с тем же параметром длиной 160 бит. Алгоритм в основном используется в средах с ограниченными ресурсами. Использование короткого ключа может сократить использование ресурсов и ускорить шифрование. В Европе многие IT-решения на основе смарт-карт используют шифрование ECC, например, национальное удостоверение личности стран Европейской экономической зоны, карта медицинского страхования в Германии и т. д. На основе эллиптических кривых созданы и работают в настоящее время криптосистемы, использующие протоколы Эль-Гамаля, RSA и Шнорра. Привлекательность подхода на основе эллиптических кривых по сравнению с классической системой RSA заключается в том, что упрощается программно-аппаратная реализация криптосистем.

Шифр Вернама или одноразовый блокнот – классический пример поточного шифра. Если для гаммы последовательность битов выбрана случайным образом и длина гаммы хотя бы равна длине сообщения, то взломать шифр невозможно. Однако этот режим шифрования имеет проблемы с передачей и хранением ключей. Поэтому основная идея современных потоковых шифров заключается в реализации концепции одноразового блокнота с использованием секретного ключа меньшей длины, из которого для гаммы генерируется псевдослучайная числовая последовательность, аналогичная случайной. Безопасность системы полностью зависит от свойств генератора потока ключей. Если генератор ключевого потока выдает повторяющийся, например, 16-битный шаблон, криптографическая стойкость системы будет незначительной.

Алгоритм хэширования MD4 был разработан Р. Ривестом в 1990 году. Описание алгоритма и пример реализации можно найти в RFC 132024 [200]. Это один из алгоритмов всего семейства MD (алгоритмы MD2 и MD5). Алгоритм построения цифрового дайджеста сообщения MD4 прост в реализации и обеспечивает построение «отпечатка пальца» для

сообщения произвольной длины. В настоящее время алгоритм используется относительно редко из-за его низкой криптостойкости. Алгоритм MD5 более надежен [49].

Алгоритмы хэширования семейства SHA – не менее известное семейство алгоритмов хэширования [49]. Как и MD5, SHA являются улучшенным продолжением MD4. Первоначально алгоритм был разработан Национальным институтом стандартов и технологий США для использования в сочетании со стандартом DSS EDS. Как и алгоритмы MD, хэширование на основе SHA включает выполнение предварительных преобразований. Б. Шнайер делает следующий вывод: алгоритм SHA, по сути, совпадает с алгоритмом MD4, отличаясь наличием расширяющего преобразования, дополнительного цикла обработки и улучшенного лавинного эффекта [201].

DSA (алгоритм цифровой подписи) использовался в утвержденном стандарте DSS в 1991 году Американским институтом стандартов (NIST). В качестве параметров алгоритм использует следующие параметры: p – простое число длиной от 64 до 1024 бит (число должно быть кратно 64); q – 160-битный простой делитель числа $p-1$. В алгоритме подписывается хэш $h(M)$ сообщения M, но не само сообщение M. Алгоритм SHA используется для хэширования сообщений. После того, как ключи сгенерированы, их владелец может подписывать свои сообщения [49].

Схема подписи Эль-Гамаля может использоваться как для шифрования, так и для цифровых подписей [49]. По сравнению, например, с ЭЦП на основе RSA, рассматриваемая схема обеспечивает более высокую производительность. В DSS и EDS Эль-Гамаля для каждой новой подписи должно использоваться новое значение k. Если третья сторона получит два сообщения, подписанные с использованием одного и того же k, ее шансы взломать закрытый ключ отправителя значительно возрастут.

Схемы Шнорра используются для построения протокола аутентификации и цифровой подписи [49]. В этом случае подписывается не само сообщение M, как в алгоритме DSA, а его хэш-функция.

Приложение 2. Программная реализация алгоритмов

Программа А2.1. Шифрование для примера 2.2.2 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the plaintext;
   m = 4; // This is the order of the quasigroup
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   label 1,2,3;
begin
Writeln('Let us carry out the encryption procedure');
Write('Enter the leader element: ');
Readln(l);
1:if  (l<>0) and (l<>1) and (l<>2) and (l<>3) then
begin
Write('Please enter leader element correctly: ');
Readln(l);
goto 1;
end;
Write('Enter ',1,' plaintext element: ');
Readln(u1);
2:if  (u1<>0) and (u1<>1) and (u1<>2) and (u1<>3) then
begin
Write('Please enter ',1,' plaintext  correctly: ');
Readln(u1);
goto 2;
end;
v1:=0;
if (l=0) and (u1=0) then  v1:=1;
if (l=0) and (u1=1) then  v1:=0;
if (l=0) and (u1=2) then  v1:=3;
if (l=0) and (u1=3) then  v1:=2;
if (l=1) and(u1=0) then  v1:=0;
if (l=1) and(u1=1) then  v1:=2;
if (l=1) and(u1=2) then  v1:=1;
if (l=1) and(u1=3) then  v1:=3;
if (l=2) and (u1=0) then  v1:=3;
if (l=2) and(u1=1) then  v1:=1;
if (l=2) and(u1=2) then  v1:=2;
if (l=2) and(u1=3) then  v1:=0;
if (l=3) and (u1=0) then  v1:=2;
if (l=3) and (u1=1) then  v1:=1;
if (l=3) and (u1=2) then  v1:=0;
if (l=3) and (u1=3) then  v1:=3;
A[1]:=u1;
B[1]:=v1;
for k:=2 to n do
begin
Write('Enter ',k,' plaintext element: ');
Readln(A[k]);
3:if  (A[k]<>0) and (A[k]<>1) and (A[k]<>2) and (A[k]<>3) then
begin
```

```
Write('Please enter ',k,' plaintext element correctly: ');
Readln(A[k]);
goto 3;
end;
if  (B[k-1]=0)and (A[k]=0) then B[k]:=1;
if  (B[k-1]=0)and (A[k]=1) then B[k]:=0;
if  (B[k-1]=0)and (A[k]=2) then B[k]:=3;
if  (B[k-1]=0)and (A[k]=3) then B[k]:=2;
if  (B[k-1]=1)and (A[k]=0) then B[k]:=0;
if  (B[k-1]=1)and (A[k]=1) then B[k]:=2;
if  (B[k-1]=1)and (A[k]=2) then B[k]:=1;
if  (B[k-1]=1)and (A[k]=3) then B[k]:=3;
if  (B[k-1]=2)and (A[k]=0) then B[k]:=3;
if  (B[k-1]=2)and (A[k]=1) then B[k]:=1;
if  (B[k-1]=2)and (A[k]=2) then B[k]:=2;
if  (B[k-1]=2)and (A[k]=3) then B[k]:=0;
if  (B[k-1]=3)and (A[k]=0) then B[k]:=2;
if  (B[k-1]=3)and (A[k]=1) then B[k]:=3;
if  (B[k-1]=3)and (A[k]=2) then B[k]:=0;
if  (B[k-1]=3)and (A[k]=3) then B[k]:=1;
end;
begin
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
    writeln;
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
 end;
end.
```

Программа А2.1. Шифрование для примера 2.2.2 (без проверки правильности ввода данных)

```
const
   n = 9; // This is the length of the plaintext;
   m = 4; // This is the order of the quasigroup
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   begin
Writeln('Let us carry out the encryption procedure');
Write('Enter the leader element: ');
Readln(l);
Write('Enter ',1,' plaintext element: ');
Readln(u1);
v1:=0;
if (l=0) and (u1=0) then  v1:=1;
if (l=0) and (u1=1) then  v1:=0;
if (l=0) and (u1=2) then  v1:=3;
if (l=0) and (u1=3) then  v1:=2;

if (l=1) and(u1=0) then  v1:=0;
if (l=1) and(u1=1) then  v1:=2;
if (l=1) and(u1=2) then  v1:=1;
if (l=1) and(u1=3) then  v1:=3;
```

```
if (l=2) and (u1=0) then  v1:=3;
if (l=2) and(u1=1) then  v1:=1;
if (l=2) and(u1=2) then  v1:=2;
if (l=2) and(u1=3) then  v1:=0;
if (l=3) and (u1=0) then  v1:=2;
if (l=3) and (u1=1) then  v1:=1;
if (l=3) and (u1=2) then  v1:=0;
if (l=3) and (u1=3) then  v1:=3;
A[1]:=u1;
B[1]:=v1;
for k:=2 to n do
begin
Write('Enter ',k,' plaintext element: ');
Readln(A[k]);
if  (B[k-1]=0)and (A[k]=0) then B[k]:=1;
if  (B[k-1]=0)and (A[k]=1) then B[k]:=0;
if  (B[k-1]=0)and (A[k]=2) then B[k]:=3;
if  (B[k-1]=0)and (A[k]=3) then B[k]:=2;
if  (B[k-1]=1)and (A[k]=0) then B[k]:=0;
if  (B[k-1]=1)and (A[k]=1) then B[k]:=2;
if  (B[k-1]=1)and (A[k]=2) then B[k]:=1;
if  (B[k-1]=1)and (A[k]=3) then B[k]:=3;
if  (B[k-1]=2)and (A[k]=0) then B[k]:=3;
if  (B[k-1]=2)and (A[k]=1) then B[k]:=1;
if  (B[k-1]=2)and (A[k]=2) then B[k]:=2;
if  (B[k-1]=2)and (A[k]=3) then B[k]:=0;
if  (B[k-1]=3)and (A[k]=0) then B[k]:=2;
if  (B[k-1]=3)and (A[k]=1) then B[k]:=3;
if  (B[k-1]=3)and (A[k]=2) then B[k]:=0;
if  (B[k-1]=3)and (A[k]=3) then B[k]:=1;
end;
begin
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
    writeln;
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
 end;
end.
```

Программа А2.2. Дешифрование для примера 2.2.2 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the ciphertext;
   m = 4; // This is the order of the quasigroup
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   label 1,2,3;
begin
Writeln('Let us carry out the decryption procedure');
Write('Enter the leader element: ');
Readln(l);
1:if  (l<>0) and (l<>1) and (l<>2) and (l<>3) then
```

```
begin
Write('Please enter leader element correctly: ');
Readln(l);
goto 1;
end;
Write('Enter ',1,' ciphertext element: ');
Readln(v1);
2:if  (v1<>0) and (v1<>1) and (v1<>2) and (v1<>3) then
begin
Write('Please enter ',1,' ciphertext  correctly: ');
Readln(v1);
goto 2;
end;
u1:=0;
if (l=0) and (v1=0) then  u1:=1;
if (l=0) and (v1=1) then  u1:=0;
if (l=0) and (v1=2) then  u1:=3;
if (l=0) and (v1=3) then  u1:=2;
if (l=1) and(v1=0) then  u1:=0;
if (l=1) and (v1=1) then  u1:=2;
if (l=1) and (v1=2) then  u1:=1;
if (l=1) and (v1=3) then  u1:=3;
if (l=2) and (v1=0) then  u1:=3;
if (l=2) and (v1=1) then  u1:=1;
if (l=2) and (v1=2) then  u1:=2;
if (l=2) and (v1=3) then  u1:=0;
if (l=3) and (v1=0) then  u1:=2;
if (l=3) and (v1=1) then  u1:=3;
if (l=3) and (v1=2) then  u1:=0;
if (l=3) and (v1=3) then  u1:=1;
A[1]:=u1;
B[1]:=v1;
for k:=2 to n do
begin
Write('Enter ',k,' ciphertext element: ');
Readln(B[k]);
3:if  (B[k]<>0) and (B[k]<>1) and (B[k]<>2) and (B[k]<>3) then
begin
Write('Please enter ',k,' ciphertext element correctly: ');
Readln(B[k]);
goto 3;
end;
if  (B[k-1]=0)and (B[k]=0) then A[k]:=1;
if  (B[k-1]=0)and (B[k]=1) then A[k]:=0;
if  (B[k-1]=0)and (B[k]=2) then A[k]:=3;
if  (B[k-1]=0)and (B[k]=3) then A[k]:=2;
if  (B[k-1]=1)and (B[k]=0) then A[k]:=0;
if  (B[k-1]=1)and (B[k]=1) then A[k]:=2;
if  (B[k-1]=1)and (B[k]=2) then A[k]:=1;
if  (B[k-1]=1)and (B[k]=3) then A[k]:=3;
if  (B[k-1]=2)and (B[k]=0) then A[k]:=3;
if  (B[k-1]=2)and (B[k]=1) then A[k]:=1;
if  (B[k-1]=2)and (B[k]=2) then A[k]:=2;
if  (B[k-1]=2)and (B[k]=3) then A[k]:=0;
if  (B[k-1]=3)and (B[k]=0) then A[k]:=2;
if  (B[k-1]=3)and (B[k]=1) then A[k]:=3;
if  (B[k-1]=3)and (B[k]=2) then A[k]:=0;
if  (B[k-1]=3)and (B[k]=3) then A[k]:=1;
end;
begin
```

```
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
    writeln;
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
end;
end.
```

Программа А2.3. Шифрование для примера 2.2.3 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the plaintext;
   m = 4; // This is the order of the left quasigroup
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
    A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   label 1,2,3;
begin
Writeln('Let us carry out the encryption procedure');
Write('Enter the leader element: ');
Readln(l);
1:if  (l<>0) and (l<>1) and (l<>2) and (l<>3) then
begin
Write('Please enter leader element correctly: ');
Readln(l);
goto 1;
end;
Write('Enter ',1,' plaintext element: ');
Readln(u1);
2:if  (u1<>0) and (u1<>1) and (u1<>2) and (u1<>3) then
begin
Write('Please enter ',1,' plaintext  correctly: ');
Readln(u1);
goto 2;
end;
v1:=0;
if (l=0) and (u1=0)then  v1:=1;
if  (l=0)and(u1=1) then  v1:=2;
if (l=0) and(u1=2) then  v1:=0;
if (l=0) and(u1=3) then  v1:=3;
if (l=1) and(u1=0) then  v1:=0;
if (l=1) and(u1=1) then  v1:=3;
if (l=1) and(u1=2) then  v1:=2;
if (l=1) and(u1=3) then  v1:=1;
if (l=2) and (u1=0)then  v1:=1;
if (l=2) and(u1=1) then  v1:=2;
if (l=2) and(u1=2) then  v1:=3;
if (l=2) and(u1=3) then  v1:=0;
if (l=3) and (u1=0) then  v1:=2;
if (l=3) and (u1=1) then  v1:=1;
if (l=3) and (u1=2) then  v1:=0;
if (l=3) and (u1=3) then  v1:=3;
A[1]:=u1;
B[1]:=v1;
for k:=2 to n do
```

```
begin
Write('Enter ',k,' plaintext element: ');
Readln(A[k]);
3:if  (A[k]<>0) and (A[k]<>1) and (A[k]<>2) and (A[k]<>3) then
begin
Write('Please enter ',k,' plaintext element correctly: ');
Readln(A[k]);
goto 3;
end;
if  (B[k-1]=0)and (A[k]=0) then B[k]:=1;
if  (B[k-1]=0)and (A[k]=1) then B[k]:=2;
if  (B[k-1]=0)and (A[k]=2) then B[k]:=0;
if  (B[k-1]=0)and (A[k]=3) then B[k]:=3;
if  (B[k-1]=1)and (A[k]=0) then B[k]:=0;
if  (B[k-1]=1)and (A[k]=1) then B[k]:=3;
if  (B[k-1]=1)and (A[k]=2) then B[k]:=2;
if  (B[k-1]=1)and (A[k]=3) then B[k]:=1;
if  (B[k-1]=2)and (A[k]=0) then B[k]:=1;
if  (B[k-1]=2)and (A[k]=1) then B[k]:=2;
if  (B[k-1]=2)and (A[k]=2) then B[k]:=3;
if  (B[k-1]=2)and (A[k]=3) then B[k]:=0;
if  (B[k-1]=3)and (A[k]=0) then B[k]:=2;
if  (B[k-1]=3)and (A[k]=1) then B[k]:=1;
if  (B[k-1]=3)and (A[k]=2) then B[k]:=0;
if  (B[k-1]=3)and (A[k]=3) then B[k]:=3;
end;
begin
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
    writeln;
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
 end;
end.
```

Программа А2.4. Дешифрование для примера 2.2.3 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the ciphertext;
   m = 4; // This is the order of the quasigroup
type
  Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   label 1,2,3;
begin
Writeln('Let us carry out the decryption procedure');
Write('Enter the leader element: ');
Readln(l);
1:if  (l<>0) and (l<>1) and (l<>2) and (l<>3) then
begin
Write('Please enter leader element correctly: ');
Readln(l);
goto 1;
end;
Write('Enter ',1,' ciphertext element: ');
```

```
Readln(v1);
2:if  (v1<>0) and (v1<>1) and (v1<>2) and (v1<>3) then
begin
Write('Please enter ',1,' ciphertext  correctly: ');
Readln(v1);
goto 2;
end;
u1:=0;
if  (l=0) and (v1=0) then  u1:=2;
if  (l=0) and (v1=1) then  u1:=0;
if (l=0) and (v1=2) then  u1:=1;
if (l=0) and (v1=3) then  u1:=3;
if  (l=1) and(v1=0) then  u1:=0;
if (l=1) and (v1=1) then  u1:=3;
if (l=1) and (v1=2) then  u1:=2;
if (l=1) and (v1=3) then  u1:=1;
if  (l=2) and (v1=0) then  u1:=3;
if  (l=2) and (v1=1) then  u1:=0;
if (l=2) and (v1=2) then  u1:=1;
if (l=2) and (v1=3) then  u1:=2;
if  (l=3) and (v1=0) then  u1:=2;
if  (l=3) and (v1=1) then  u1:=1;
if (l=3) and (v1=2) then  u1:=0;
if (l=3) and (v1=3) then  u1:=3;
A[1]:=u1;
B[1]:=v1;
for k:=2 to n do
begin
Write('Enter ',k,' ciphertext element: ');
Readln(B[k]);
3:if  (B[k]<>0) and (B[k]<>1) and (B[k]<>2) and (B[k]<>3) then
begin
Write('Please enter ',k,' ciphertext element correctly: ');
Readln(B[k]);
goto 3;
end;
if  (B[k-1]=0)and (B[k]=0) then A[k]:=2;
if  (B[k-1]=0)and (B[k]=1) then A[k]:=0;
if  (B[k-1]=0)and (B[k]=2) then A[k]:=1;
if  (B[k-1]=0)and (B[k]=3) then A[k]:=3;
if  (B[k-1]=1)and (B[k]=0) then A[k]:=0;
if  (B[k-1]=1)and (B[k]=1) then A[k]:=3;
if  (B[k-1]=1)and (B[k]=2) then A[k]:=2;
if  (B[k-1]=1)and (B[k]=3) then A[k]:=1;
if  (B[k-1]=2)and (B[k]=0) then A[k]:=3;
if  (B[k-1]=2)and (B[k]=1) then A[k]:=0;
if  (B[k-1]=2)and (B[k]=2) then A[k]:=1;
if  (B[k-1]=2)and (B[k]=3) then A[k]:=2;
if  (B[k-1]=3)and (B[k]=0) then A[k]:=2;
if  (B[k-1]=3)and (B[k]=1) then A[k]:=1;
if  (B[k-1]=3)and (B[k]=2) then A[k]:=0;
if  (B[k-1]=3)and (B[k]=3) then A[k]:=3;
end;
begin
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
    writeln;
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
end;
end.
```

Программа А2.5. Шифрование для примера 2.3.1 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the plaintext;
   m = 5; // This is the order of the right quasigroup
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   label 1,2,3;
begin
Writeln('Let us carry out the encryption procedure');
Write('Enter the leader element: ');
Readln(l);
1:if  (l<>0) and (l<>1) and (l<>2) and (l<>3) and (l<>4)then
begin
Write('Please enter leader element correctly: ');
Readln(l);
goto 1;
end;
Write('Enter ',1,' plaintext element: ');
Readln(u1);
2:if  (u1<>0) and (u1<>1) and (u1<>2) and (u1<>3) and (u1<>4)then
begin
Write('Please enter ',1,' plaintext  correctly: ');
Readln(u1);
goto 2;
end;
v1:=0;
if (l=0) and(u1=0) then  v1:=0;
if (l=0) and(u1=1) then  v1:=2;
if (l=0) and(u1=2) then  v1:=3;
if (l=0) and(u1=3) then  v1:=1;
if (l=0) and(u1=4) then  v1:=4;
if (l=1) and(u1=0) then  v1:=3;
if (l=1) and(u1=1) then  v1:=0;
if (l=1) and(u1=2) then  v1:=2;
if (l=1) and(u1=3) then  v1:=1;
if (l=1) and(u1=4) then  v1:=4;
if (l=2) and(u1=0) then  v1:=2;
if (l=2) and(u1=1) then  v1:=1;
if (l=2) and(u1=2) then  v1:=0;
if (l=2) and(u1=3) then  v1:=4;
if (l=2) and(u1=4) then  v1:=3;
if (l=3) and (u1=0) then  v1:=1;
if (l=3) and (u1=1) then  v1:=0;
if (l=3) and (u1=2) then  v1:=2;
if (l=3) and (u1=3) then  v1:=3;
if (l=3) and(u1=4) then  v1:=4;
if  (l=4) and (u1=0) then  v1:=0;
if  (l=4) and (u1=1) then  v1:=4;
if (l=4) and (u1=2) then  v1:=1;
if (l=4) and (u1=3) then  v1:=2;
if (l=4) and(u1=4) then  v1:=3;
A[1]:=u1;
```

```
B[1]:=v1;
for k:=2 to n do
begin
Write('Enter ',k,' plaintext element: ');
Readln(A[k]);
3:if  (A[k]<>0) and (A[k]<>1) and (A[k]<>2) and (A[k]<>3) and (A[k]<>4)then
begin
Write('Please enter ',k,' plaintext element correctly: ');
Readln(A[k]);
goto 3;
end;
if  (A[k]=0)and (B[k-1]=0) then B[k]:=0;
if  (A[k]=0)and (B[k-1]=1) then B[k]:=3;
if  (A[k]=0)and (B[k-1]=2) then B[k]:=2;
if  (A[k]=0)and (B[k-1]=3) then B[k]:=1;
if  (A[k]=0)and (B[k-1]=4) then B[k]:=0;
if  (A[k]=1)and (B[k-1]=0) then B[k]:=2;
if  (A[k]=1)and (B[k-1]=1) then B[k]:=0;
if  (A[k]=1)and (B[k-1]=2) then B[k]:=1;
if  (A[k]=1)and (B[k-1]=3) then B[k]:=0;
if  (A[k]=1)and (B[k-1]=4) then B[k]:=4;
if  (A[k]=2)and (B[k-1]=0) then B[k]:=3;
if  (A[k]=2)and (B[k-1]=1) then B[k]:=2;
if  (A[k]=2)and (B[k-1]=2) then B[k]:=0;
if  (A[k]=2)and (B[k-1]=3) then B[k]:=2;
if  (A[k]=2)and (B[k-1]=4) then B[k]:=1;
if  (A[k]=3)and (B[k-1]=0) then B[k]:=1;
if  (A[k]=3)and (B[k-1]=1) then B[k]:=1;
if  (A[k]=3)and (B[k-1]=2) then B[k]:=4;
if  (A[k]=3)and (B[k-1]=3) then B[k]:=3;
if  (A[k]=3)and (B[k-1]=4) then B[k]:=2;
if  (A[k]=4)and (B[k-1]=0) then B[k]:=4;
if  (A[k]=4)and (B[k-1]=1) then B[k]:=4;
if  (A[k]=4)and (B[k-1]=2) then B[k]:=3;
if  (A[k]=4)and (B[k-1]=3) then B[k]:=4;
if  (A[k]=4)and (B[k-1]=4) then B[k]:=3;
end;
begin
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
    writeln;
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
 end;
      end.
```

Программа А2.6. Дешифрование для примера 2.3.1 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the ciphertext;
   m = 5; // This is the order of the right quasigroup
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B: mas;
   u1, v1,l:Date_u1;
   i,k: Date_n;
   label 1,2,3;
```

```
begin
Writeln('Let us carry out the decryption procedure');
Write('Enter the leader element: ');
Readln(l);
1:if  (l<>0) and (l<>1) and (l<>2) and (l<>3) and (l<>4)then
begin
Write('Please enter leader element correctly: ');
Readln(l);
goto 1;
end;
Write('Enter ',1,' ciphertext element: ');
Readln(v1);
2:if  (v1<>0) and (v1<>1) and (v1<>2) and (v1<>3)and (v1<>4) then
begin
Write('Please enter ',1,' ciphertext  correctly: ');
Readln(v1);
goto 2;
end;
u1:=0;
if (l=0) and  (v1=0) then  u1:=0;
if (l=0) and (v1=1) then  u1:=3;
if (l=0) and (v1=2) then  u1:=1;
if (l=0) and (v1=3) then  u1:=2;
if (l=0) and (v1=4) then  u1:=4;
if (l=1) and(v1=0) then  u1:=1;
if (l=1) and (v1=1) then  u1:=3;
if (l=1) and(v1=2) then  u1:=2;
if (l=1) and(v1=3) then  u1:=0;
if (l=1) and(v1=4) then  u1:=4;
if (l=2) and (v1=0) then  u1:=2;
if (l=2) and (v1=1) then  u1:=1;
if (l=2) and (v1=2) then  u1:=0;
if (l=2) and (v1=3) then  u1:=4;
if (l=2) and (v1=4) then  u1:=3;
if (l=3) and (v1=0) then  u1:=1;
if (l=3) and  (v1=1) then  u1:=0;
if (l=3) and (v1=2) then  u1:=2;
if (l=3) and (v1=3) then  u1:=3;
if (l=3) and (v1=4) then  u1:=4;
if (l=4) and (v1=0) then  u1:=0;
if (l=4) and(v1=1) then  u1:=2;
if (l=4) and(v1=2) then  u1:=3;
if (l=4) and(v1=3) then  u1:=4;
if (l=4) and(v1=4) then  u1:=1;
A[1]:=u1;
B[1]:=v1;
for k:=2 to n do
begin
Write('Enter ',k,' ciphertext element: ');
Readln(B[k]);
3:if  (B[k]<>0) and (B[k]<>1) and (B[k]<>2) and (B[k]<>3)and (B[k]<>4) then
begin
Write('Please enter ',k,' ciphertext element correctly: ');
Readln(B[k]);
goto 3;
end;
if  (B[k-1]=0)and (B[k]=0) then A[k]:=0;
if  (B[k-1]=1)and (B[k]=0) then A[k]:=1;
if  (B[k-1]=2)and (B[k]=0) then A[k]:=2;
if  (B[k-1]=3)and (B[k]=0) then A[k]:=1;
```

```
if  (B[k-1]=4)and (B[k]=0) then A[k]:=0;
if  (B[k-1]=0)and (B[k]=1) then A[k]:=3;
if  (B[k-1]=1)and (B[k]=1) then A[k]:=3;
if  (B[k-1]=2)and (B[k]=1) then A[k]:=1;
if  (B[k-1]=3)and (B[k]=1) then A[k]:=0;
if  (B[k-1]=4)and (B[k]=1) then A[k]:=2;
if  (B[k-1]=0)and (B[k]=2) then A[k]:=1;
if  (B[k-1]=1)and (B[k]=2) then A[k]:=2;
if  (B[k-1]=2)and (B[k]=2) then A[k]:=0;
if  (B[k-1]=3)and (B[k]=2) then A[k]:=2;
if  (B[k-1]=4)and (B[k]=2) then A[k]:=3;
if  (B[k-1]=0)and (B[k]=3) then A[k]:=2;
if  (B[k-1]=1)and (B[k]=3) then A[k]:=0;
if  (B[k-1]=2)and (B[k]=3) then A[k]:=4;
if  (B[k-1]=3)and (B[k]=3) then A[k]:=3;
if  (B[k-1]=4)and (B[k]=3) then A[k]:=4;
if  (B[k-1]=0)and (B[k]=4) then A[k]:=4;
if  (B[k-1]=1)and (B[k]=4) then A[k]:=4;
if  (B[k-1]=2)and (B[k]=4) then A[k]:=3;
if  (B[k-1]=3)and (B[k]=4) then A[k]:=4;
if  (B[k-1]=4)and (B[k]=4) then A[k]:=1;
end;
begin
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
    writeln;
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
end;
      end.
```

Таблица А2.1. Выводы по программам А2.1.-А2.6. (processor: Intel (R) Core (TM) i3-8130 U CPU @ 220GHz)

Порядок квазигруппы m	**Длина текста n**	**Используемый алгоритм**	**Лидер**	**Открытый текст U**	**Зашифрованный текст V**	**Средняя скорость обработки (ms)**		**Оценка сложности алгоритма $O(f(n))$**
						без проверки ввода данных	**с проверкой**	
$m = 4$	$n = 9$	Алгоритм Марковского для шифрования	$l = 3$	031323110	200220001	75.2115	83,7713	линейная $O(n)$
$m = 4$	$n = 9$	Алгоритм Марковского для дешифрования	$l = 3$	031323110	200220001	72.0278	76,0129	линейная $O(n)$
$m = 4$ (левая квазигруппа)	$n = 9$	Алгоритм Марковского для шифрования	$l = 3$	031323110	202003132	59.8775	61,3143	линейная $O(n)$
$m = 4$ (левая квазигруппа)	$n = 9$	Алгоритм Марковского для дешифрования	$l = 3$	031323110	202003132	67.4899	77,0291	линейная $O(n)$
$m = 5$ (правая квазигруппа)	$n = 9$	Алгоритм Марковского для шифрования	$l = 3$	031323110	244201022	73.5281	79,691	линейная $O(n)$
$m = 5$ (правая квазигруппа)	$n = 9$	Алгоритм Марковского для дешифрования	$l = 3$	031323110	244201022	70.7163	74,1569	линейная $O(n)$

Программа А2.7. Шифрование для примера 2.4.6 (с проверкой правильности ввода данных)

```
const
    n = 9; // This is the length of the plaintext;
   m = 3; // This is the order of the groupoid
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B : mas;
   u1,v1,u2,v2,l1,l2,l3,l4:Date_u1;
   i,k: Date_n;
   label 1,2,3,4,5,6,7;
begin
Writeln ('Let us start the encryption procedure:');
Write('Enter 1 leader element: ');
Readln(l1);
1:if  (l1<>0) and (l1<>1) and (l1<>2) then
begin
Write('Please enter ',1,' leader element correctly: ');
Readln(l1);
goto 1;
end;
Write('Enter 2 leader element: ');
Readln(l2);
2:if  (l2<>0) and (l2<>1) and (l2<>2) then
begin
Write('Please enter ',2,' leader element correctly: ');
Readln(l2);
goto 2;
end;
Write('Enter 3 leader element: ');
Readln(l3);
3:if  (l3<>0) and (l3<>1) and (l3<>2) then
begin
Write('Please enter ',3,' leader element correctly: ');
Readln(l3);
goto 3;
end;
Write('Enter 4 leader element: ');
Readln(l4);
4:if  (l4<>0) and (l4<>1) and (l4<>2) then
begin
Write('Please enter ',4,' leader element correctly: ');
Readln(l4);
goto 4;
end;
Write ('Enter ',1,' plaintext element: ');
Readln(u1);
5:if  (u1<>0) and (u1<>1) and (u1<>2) then
begin
Write('Please enter ',1,' plaintext  correctly: ');
Readln(u1);
goto 5;
end;
v1:=0;
if  (u1=0) and (l1=0) and (l2=0) then  v1:=0;
if  (u1=0) and (l1=0) and (l2=1) then  v1:=0;
```

```
if  (u1=0) and (l1=0) and (l2=2) then  v1:=1;
if  (u1=0) and (l1=1) and (l2=0) then  v1:=0;
if  (u1=0) and (l1=1) and (l2=1) then  v1:=0;
if  (u1=0) and (l1=1) and (l2=2) then  v1:=1;
if  (u1=0) and (l1=2) and (l2=0) then  v1:=2;
if  (u1=0) and (l1=2) and (l2=1) then  v1:=2;
if  (u1=0) and (l1=2) and (l2=2) then  v1:=0;
if  (u1=1) and (l1=0) and (l2=0) then  v1:=1;
if  (u1=1) and (l1=0) and (l2=1) then  v1:=1;
if  (u1=1) and (l1=0) and (l2=2) then  v1:=2;
if  (u1=1) and (l1=1) and (l2=0) then  v1:=1;
if  (u1=1) and (l1=1) and (l2=1) then  v1:=1;
if  (u1=1) and (l1=1) and (l2=2) then  v1:=2;
if  (u1=1) and (l1=2) and (l2=0) then  v1:=0;
if  (u1=1) and (l1=2) and (l2=1) then  v1:=0;
if  (u1=1) and (l1=2) and (l2=2) then  v1:=1;
if  (u1=2) and (l1=0) and (l2=0) then  v1:=2;
if  (u1=2) and (l1=0) and (l2=1) then  v1:=2;
if  (u1=2) and (l1=0) and (l2=2) then  v1:=0;
if  (u1=2) and (l1=1) and (l2=0) then  v1:=2;
if  (u1=2) and (l1=1) and (l2=1) then  v1:=2;
if  (u1=2) and (l1=1) and (l2=2) then  v1:=0;
if  (u1=2) and (l1=2) and (l2=0) then  v1:=1;
if  (u1=2) and (l1=2) and (l2=1) then  v1:=1;
if  (u1=2) and (l1=2) and (l2=2) then  v1:=2;
write ('Enter ',2,' plaintext element: ');
Readln(u2);
6:if  (u2<>0) and (u2<>1) and (u2<>2) then
begin
Write('Please enter ',2,' plaintext  correctly: ');
Readln(u2);
goto 6;
end;
v2:=0;
if  (u2=0) and (l3=0) and (l4=0) then  v2:=0;
if  (u2=0) and (l3=0) and (l4=1) then  v2:=0;
if  (u2=0) and (l3=0) and (l4=2) then  v2:=1;
if  (u2=0) and (l3=1) and (l4=0) then  v2:=0;
if  (u2=0) and (l3=1) and (l4=1) then  v2:=0;
if  (u2=0) and (l3=1) and (l4=2) then  v2:=1;
if  (u2=0) and (l3=2) and (l4=0) then  v2:=2;
if  (u2=0) and (l3=2) and (l4=1) then  v2:=2;
if  (u2=0) and (l3=2) and (l4=2) then  v2:=0;
if  (u2=1) and (l3=0) and (l4=0) then  v2:=1;
if  (u2=1) and (l3=0) and (l4=1) then  v2:=1;
if  (u2=1) and (l3=0) and (l4=2) then  v2:=2;
if  (u2=1) and (l3=1) and (l4=0) then  v2:=1;
if  (u2=1) and (l3=1) and (l4=1) then  v2:=1;
if  (u2=1) and (l3=1) and (l4=2) then  v2:=2;
if  (u2=1) and (l3=2) and (l4=0) then  v2:=0;
if  (u2=1) and (l3=2) and (l4=1) then  v2:=0;
if  (u2=1) and (l3=2) and (l4=2) then  v2:=1;
if  (u2=2) and (l3=0) and (l4=0) then  v2:=2;
if  (u2=2) and (l3=0) and (l4=1) then  v2:=2;
if  (u2=2) and (l3=0) and (l4=2) then  v2:=0;
if  (u2=2) and (l3=1) and (l4=0) then  v2:=2;
if  (u2=2) and (l3=1) and (l4=1) then  v2:=2;
if  (u2=2) and (l3=1) and (l4=2) then  v2:=0;
if  (u2=2) and (l3=2) and (l4=0) then  v2:=1;
if  (u2=2) and (l3=2) and (l4=1) then  v2:=1;
```

```
if  (u2=2) and (l3=2) and (l4=2) then  v2:=2;
B[1]:=v1;
B[2]:=v2;
A[1]:=u1;
A[2]:=u2;
for k:=3 to n do
begin
Write('Enter ',k,' plaintext element: ');
Readln(A[k]);
7:if  (A[k]<>0) and (A[k]<>1) and (A[k]<>2) then
begin
Write('Please enter ',k,' plaintext element correctly: ');
Readln(A[k]);
goto 7;
end;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=0) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=1) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=2) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=0) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=1) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=2) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=0) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=1) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=2) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=0) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=1) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=2) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=2) then B[k]:=2;
   end;
begin
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
    writeln;
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
 end;
      end.
```

Программа А2. 8. Дешифрование для примера 2.4.6 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the ciphertext;
   m = 3; // This is the order of the groupoid
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
```

```
   mas = array[1..n] of integer;
var
   A,B : mas;
   u1,v1,u2,v2,l1,l2,l3,l4:Date_u1;
   i,k: Date_n;
   label 1,2,3,4,5,6,7;
begin
   Writeln ('Let us start the decryption procedure:');
Write('Enter 1 leader element: ');
Readln(l1);
1:if  (l1<>0) and (l1<>1) and (l1<>2) then
begin
Write('Please enter ',1,' leader element correctly: ');
Readln(l1);
goto 1;
end;
Write('Enter 2 leader element: ');
Readln(l2);
2:if  (l2<>0) and (l2<>1) and (l2<>2) then
begin
Write('Please enter ',2,' leader element correctly: ');
Readln(l2);
goto 2;
end;
Write('Enter 3 leader element: ');
Readln(l3);
3:if  (l3<>0) and (l3<>1) and (l3<>2) then
begin
Write('Please enter ',3,' leader element correctly: ');
Readln(l3);
goto 3;
end;
Write('Enter 4 leader element: ');
Readln(l4);
4:if  (l4<>0) and (l4<>1) and (l4<>2) then
begin
Write('Please enter ',4,' leader element correctly: ');
Readln(l4);
goto 4;
end;
Write ('Enter ',1,' ciphertext element: ');
Readln(v1);
5:if  (v1<>0) and (v1<>1) and (v1<>2) then
begin
Write('Please enter ',1,' ciphertext  correctly: ');
Readln(v1);
goto 5;
end;
u1:=0;
if  (v1=0) and (l1=0) and (l2=0) then  u1:=0;
if  (v1=0) and (l1=0) and (l2=1) then  u1:=0;
if  (v1=0) and (l1=0) and (l2=2) then  u1:=2;
if  (v1=0) and (l1=1) and (l2=0) then  u1:=0;
if  (v1=0) and (l1=1) and (l2=1) then  u1:=0;
if  (v1=0) and (l1=1) and (l2=2) then  u1:=2;
if  (v1=0) and (l1=2) and (l2=0) then  u1:=1;
if  (v1=0) and (l1=2) and (l2=1) then  u1:=1;
if  (v1=0) and (l1=2) and (l2=2) then  u1:=0;
if  (v1=1) and (l1=0) and (l2=0) then  u1:=1;
if  (v1=1) and (l1=0) and (l2=1) then  u1:=1;
```

```
if  (v1=1) and (l1=0) and (l2=2) then  u1:=0;
if  (v1=1) and (l1=1) and (l2=0) then  u1:=1;
if  (v1=1) and (l1=1) and (l2=1) then  u1:=1;
if  (v1=1) and (l1=1) and (l2=2) then  u1:=0;
if  (v1=1) and (l1=2) and (l2=0) then  u1:=2;
if  (v1=1) and (l1=2) and (l2=1) then  u1:=2;
if  (v1=1) and (l1=2) and (l2=2) then  u1:=1;
if  (v1=2) and (l1=0) and (l2=0) then  u1:=2;
if  (v1=2) and (l1=0) and (l2=1) then  u1:=2;
if  (v1=2) and (l1=0) and (l2=2) then  u1:=1;
if  (v1=2) and (l1=1) and (l2=0) then  u1:=2;
if  (v1=2) and (l1=1) and (l2=1) then  u1:=2;
if  (v1=2) and (l1=1) and (l2=2) then  u1:=1;
if  (v1=2) and (l1=2) and (l2=0) then  u1:=0;
if  (v1=2) and (l1=2) and (l2=1) then  u1:=0;
if  (v1=2) and (l1=2) and (l2=2) then  u1:=2;
write ('Enter ',2,' ciphertext element: ');
Readln(v2);
6:if  (v2<>0) and (v2<>1) and (v2<>2) then
begin
Write('Please enter ',2,' ciphertext  correctly: ');
Readln(v2);
goto 6;
end;
u2:=0;
if  (v2=0) and (l3=0) and (l4=0) then  u2:=0;
if  (v2=0) and (l3=0) and (l4=1) then  u2:=0;
if  (v2=0) and (l3=0) and (l4=2) then  u2:=2;
if  (v2=0) and (l3=1) and (l4=0) then  u2:=0;
if  (v2=0) and (l3=1) and (l4=1) then  u2:=0;
if  (v2=0) and (l3=1) and (l4=2) then  u2:=2;
if  (v2=0) and (l3=2) and (l4=0) then  u2:=1;
if  (v2=0) and (l3=2) and (l4=1) then  u2:=1;
if  (v2=0) and (l3=2) and (l4=2) then  u2:=0;
if  (v2=1) and (l3=0) and (l4=0) then  u2:=1;
if  (v2=1) and (l3=0) and (l4=1) then  u2:=1;
if  (v2=1) and (l3=0) and (l4=2) then  u2:=0;
if  (v2=1) and (l3=1) and (l4=0) then  u2:=1;
if  (v2=1) and (l3=1) and (l4=1) then  u2:=1;
if  (v2=1) and (l3=1) and (l4=2) then  u2:=0;
if  (v2=1) and (l3=2) and (l4=0) then  u2:=2;
if  (v2=1) and (l3=2) and (l4=1) then  u2:=2;
if  (v2=1) and (l3=2) and (l4=2) then  u2:=1;
if  (v2=2) and (l3=0) and (l4=0) then  u2:=2;
if  (v2=2) and (l3=0) and (l4=1) then  u2:=2;
if  (v2=2) and (l3=0) and (l4=2) then  u2:=1;
if  (v2=2) and (l3=1) and (l4=0) then  u2:=2;
if  (v2=2) and (l3=1) and (l4=1) then  u2:=2;
if  (v2=2) and (l3=1) and (l4=2) then  u2:=1;
if  (v2=2) and (l3=2) and (l4=0) then  u2:=0;
if  (v2=2) and (l3=2) and (l4=1) then  u2:=0;
if  (v2=2) and (l3=2) and (l4=2) then  u2:=2;
B[1]:=v1;
B[2]:=v2;
A[1]:=u1;
A[2]:=u2;
for k:=3 to n do
begin
Write('Enter ',k,' ciphertext element: ');
Readln(B[k]);
```

```
7:if  (B[k]<>0) and (B[k]<>1) and (B[k]<>2) then
begin
Write('Please enter ',k,' plaintext element correctly: ');
Readln(B[k]);
goto 7;
end;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=0) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=1) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=2) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=0) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=1) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=2) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=0) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=1) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=2) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=0) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=1) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=2) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=2) then A[k]:=2;
   end;
begin
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
    writeln;
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
end;
      end.
```

Программа А2. 9. Шифрование для примера 2.5.3 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the plaintext;
   m = 3; // This is the order of the groupoid
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B : mas;
   u1,v1,u2,v2,l1,l2,l3,l4,t2,t4:Date_u1;
   i,k: Date_n;
   label 1,2,3,4,5,6,7;
begin
Writeln ('Let us start the encryption procedure:');
Write('Enter 1 leader element: ');
```

```
Readln(l1);
1:if  (l1<>0) and (l1<>1) and (l1<>2) then
begin
Write('Please enter ',1,' leader element correctly: ');
Readln(l1);
goto 1;
end;
Write('Enter 2 leader element: ');
Readln(l2);
2:if  (l2<>0) and (l2<>1) and (l2<>2) then
begin
Write('Please enter ',2,' leader element correctly: ');
Readln(l2);
goto 2;
end;
Write('Enter 3 leader element: ');
Readln(l3);
3:if  (l3<>0) and (l3<>1) and (l3<>2) then
begin
Write('Please enter ',3,' leader element correctly: ');
Readln(l3);
goto 3;
end;
Write ('Enter ',1,' plaintext element: ');
Readln(u1);
5:if  (u1<>0) and (u1<>1) and (u1<>2) then
begin
Write('Please enter ',1,' plaintext  correctly: ');
Readln(u1);
goto 5;
end;
A[1]:=u1;
v1:=0;
if  (l1=0) and (l2=0) and (u1=0) then  v1:=1;
if  (l1=0) and (l2=0) and (u1=1) then  v1:=2;
if  (l1=0) and (l2=0) and (u1=2) then  v1:=0;
if  (l1=0) and (l2=1) and (u1=0) then  v1:=0;
if  (l1=0) and (l2=1) and (u1=1) then  v1:=1;
if  (l1=0) and (l2=1) and (u1=2) then  v1:=2;
if  (l1=0) and (l2=2) and (u1=0) then  v1:=2;
if  (l1=0) and (l2=2) and (u1=1) then  v1:=0;
if  (l1=0) and (l2=2) and (u1=2) then  v1:=1;
if  (l1=1) and (l2=0) and (u1=0) then  v1:=2;
if  (l1=1) and (l2=0) and (u1=1) then  v1:=0;
if  (l1=1) and (l2=0) and (u1=2) then  v1:=1;
if  (l1=1) and (l2=1) and (u1=0) then  v1:=1;
if  (l1=1) and (l2=1) and (u1=1) then  v1:=2;
if  (l1=1) and (l2=1) and (u1=2) then  v1:=0;
if  (l1=1) and (l2=2) and (u1=0) then  v1:=0;
if  (l1=1) and (l2=2) and (u1=1) then  v1:=1;
if  (l1=1) and (l2=2) and (u1=2) then  v1:=2;
if  (l1=2) and (l2=0) and (u1=0) then  v1:=0;
if  (l1=2) and (l2=0) and (u1=1) then  v1:=1;
if  (l1=2) and (l2=0) and (u1=2) then  v1:=2;
if  (l1=2) and (l2=1) and (u1=0) then  v1:=2;
if  (l1=2) and (l2=1) and (u1=1) then  v1:=0;
if  (l1=2) and (l2=1) and (u1=2) then  v1:=1;
if  (l1=2) and (l2=2) and (u1=0) then  v1:=1;
if  (l1=2) and (l2=2) and (u1=1) then  v1:=2;
if  (l1=2) and (l2=2) and (u1=2) then  v1:=0;
```

```
write ('Enter ',2,' plaintext element: ');
Readln(u2);
6:if  (u2<>0) and (u2<>1) and (u2<>2) then
begin
Write('Please enter ',2,' plaintext  correctly: ');
Readln(u2);
goto 6;
end;
A[2]:=u2;
t2:=0;
if  (l3=0) and (v1=0) and (u2=0) then  t2:=1;
if  (l3=0) and (v1=0) and (u2=1) then  t2:=2;
if  (l3=0) and (v1=0) and (u2=2) then  t2:=0;
if  (l3=0) and (v1=1) and (u2=0) then  t2:=0;
if  (l3=0) and (v1=1) and (u2=1) then  t2:=1;
if  (l3=0) and (v1=1) and (u2=2) then  t2:=2;
if  (l3=0) and (v1=2) and (u2=0) then  t2:=2;
if  (l3=0) and (v1=2) and (u2=1) then  t2:=0;
if  (l3=0) and (v1=2) and (u2=2) then  t2:=1;
if  (l3=1) and (v1=0) and (u2=0) then  t2:=2;
if  (l3=1) and (v1=0) and (u2=1) then  t2:=0;
if  (l3=1) and (v1=0) and (u2=2) then  t2:=1;
if  (l3=1) and (v1=1) and (u2=0) then  t2:=1;
if  (l3=1) and (v1=1) and (u2=1) then  t2:=2;
if  (l3=1) and (v1=1) and (u2=2) then  t2:=0;
if  (l3=1) and (v1=2) and (u2=0) then  t2:=0;
if  (l3=1) and (v1=2) and (u2=1) then  t2:=1;
if  (l3=1) and (v1=2) and (u2=2) then  t2:=2;
if  (l3=2) and (v1=0) and (u2=0) then  t2:=0;
if  (l3=2) and (v1=0) and (u2=1) then  t2:=1;
if  (l3=2) and (v1=0) and (u2=2) then  t2:=2;
if  (l3=2) and (v1=1) and (u2=0) then  t2:=2;
if  (l3=2) and (v1=1) and (u2=1) then  t2:=0;
if  (l3=2) and (v1=1) and (u2=2) then  t2:=1;
if  (l3=2) and (v1=2) and (u2=0) then  t2:=1;
if  (l3=2) and (v1=2) and (u2=1) then  t2:=2;
if  (l3=2) and (v1=2) and (u2=2) then  t2:=0;
v2:=0;
if  (l3=0) and (v1=0) and (t2=0) then  v2:=1;
if  (l3=0) and (v1=0) and (t2=1) then  v2:=2;
if  (l3=0) and (v1=0) and (t2=2) then  v2:=0;
if  (l3=0) and (v1=1) and (t2=0) then  v2:=0;
if  (l3=0) and (v1=1) and (t2=1) then  v2:=1;
if  (l3=0) and (v1=1) and (t2=2) then  v2:=2;
if  (l3=0) and (v1=2) and (t2=0) then  v2:=2;
if  (l3=0) and (v1=2) and (t2=1) then  v2:=0;
if  (l3=0) and (v1=2) and (t2=2) then  v2:=1;
if  (l3=1) and (v1=0) and (t2=0) then  v2:=2;
if  (l3=1) and (v1=0) and (t2=1) then  v2:=0;
if  (l3=1) and (v1=0) and (t2=2) then  v2:=1;
if  (l3=1) and (v1=1) and (t2=0) then  v2:=1;
if  (l3=1) and (v1=1) and (t2=1) then  v2:=2;
if  (l3=1) and (v1=1) and (t2=2) then  v2:=0;
if  (l3=1) and (v1=2) and (t2=0) then  v2:=0;
if  (l3=1) and (v1=2) and (t2=1) then  v2:=1;
if  (l3=1) and (v1=2) and (t2=2) then  v2:=2;
if  (l3=2) and (v1=0) and (t2=0) then  v2:=0;
if  (l3=2) and (v1=0) and (t2=1) then  v2:=1;
if  (l3=2) and (v1=0) and (t2=2) then  v2:=2;
if  (l3=2) and (v1=1) and (t2=0) then  v2:=2;
```

```
if  (l3=2) and (v1=1) and (t2=1) then  v2:=0;
if  (l3=2) and (v1=1) and (t2=2) then  v2:=1;
if  (l3=2) and (v1=2) and (t2=0) then  v2:=1;
if  (l3=2) and (v1=2) and (t2=1) then  v2:=2;
if  (l3=2) and (v1=2) and (t2=2) then  v2:=0;
B[1]:=v1;
B[2]:=v2;
for k:=3 to n do
begin
Write('Enter ',k,' plaintext element: ');
Readln(A[k]);
7:if  (A[k]<>0) and (A[k]<>1) and (A[k]<>2) then
begin
Write('Please enter ',k,' plaintext element correctly: ');
Readln(A[k]);
goto 7;
end;
if k mod 2=1 then
begin
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=0) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=1) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=2) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=0) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=1) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=2) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=0) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=1) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=2) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=0) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=1) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=2) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=0) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=1) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=2) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=0) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=1) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=2) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=0) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=1) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=2) then B[k]:=0;
end;
t4:=0;
if k mod 2=0 then
begin
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=0) then t4:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=1) then t4:=2;
if  (B[k-2]=0)and (B[k-1]=0) and (A[k]=2) then t4:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=0) then t4:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=1) then t4:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (A[k]=2) then t4:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=0) then t4:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=1) then t4:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (A[k]=2) then t4:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=0) then t4:=2;
```

```
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=1) then t4:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (A[k]=2) then t4:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=0) then t4:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=1) then t4:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (A[k]=2) then t4:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=0) then t4:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=1) then t4:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (A[k]=2) then t4:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=0) then t4:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=1) then t4:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (A[k]=2) then t4:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=0) then t4:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=1) then t4:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (A[k]=2) then t4:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=0) then t4:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=1) then t4:=2;
if  (B[k-2]=2)and (B[k-1]=2) and (A[k]=2) then t4:=0;
if  (B[k-2]=0)and (B[k-1]=0) and (t4=0) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (t4=1) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=0) and (t4=2) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (t4=0) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (t4=1) then B[k]:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (t4=2) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (t4=0) then B[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (t4=1) then B[k]:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (t4=2) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (t4=0) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=0) and (t4=1) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (t4=2) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (t4=0) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (t4=1) then B[k]:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (t4=2) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (t4=0) then B[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (t4=1) then B[k]:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (t4=2) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (t4=0) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (t4=1) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (t4=2) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (t4=0) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (t4=1) then B[k]:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (t4=2) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (t4=0) then B[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (t4=1) then B[k]:=2;
if  (B[k-2]=2)and (B[k-1]=2) and (t4=2) then B[k]:=0;
end;
end;
begin
    Write('Plaintext : ');
    for k := 1 to n do write( A[k]);
    writeln;
    Write('Ciphertext: ');
    for k := 1 to n do write( B[k]);
 end;
      end.
```

Программа А2.10. Дешифрование для примера 2.5.3 (с проверкой правильности ввода данных)

```
const
   n = 9; // This is the length of the ciphertext;
   m = 3; // This is the order of the groupoid
type
   Date_n = 1..n;
   Date_u1 = 0..m-1;
   mas = array[1..n] of integer;
var
   A,B : mas;
   u1,v1,u2,v2,l1,l2,l3,t1,t3:Date_u1;
   i,k: Date_n;
   label 1,2,3,4,5,6,7;
begin
Writeln ('Let us start the decryption procedure:');
Write('Enter 1 leader element: ');
Readln(l1);
1:if  (l1<>0) and (l1<>1) and (l1<>2) then
begin
Write('Please enter ',1,' leader element correctly: ');
Readln(l1);
goto 1;
end;
Write('Enter 2 leader element: ');
Readln(l2);
2:if  (l2<>0) and (l2<>1) and (l2<>2) then
begin
Write('Please enter ' ,2,' leader element correctly: ');
Readln(l2);
goto 2;
end;
Write('Enter 3 leader element: ');
Readln(l3);
3:if  (l3<>0) and (l3<>1) and (l3<>2) then
begin
Write('Please enter ', 3,' leader element correctly: ');
Readln(l3);
goto 3;
end;
Write ('Enter ',1,' chifertext element: ');
Readln(v1);
5:if  (v1<>0) and (v1<>1) and (v1<>2) then
begin
Write('Please enter ',1,' ciphertext  correctly: ');
Readln(v1);
goto 5;
end;
t1:=0;
if  (l1=0) and (l2=0) and (v1=0) then  t1:=1;
if  (l1=0) and (l2=0) and (v1=1) then  t1:=2;
if  (l1=0) and (l2=0) and (v1=2) then  t1:=0;
if  (l1=0) and (l2=1) and (v1=0) then  t1:=0;
if  (l1=0) and (l2=1) and (v1=1) then  t1:=1;
if  (l1=0) and (l2=1) and (v1=2) then  t1:=2;
if  (l1=0) and (l2=2) and (v1=0) then  t1:=2;
if  (l1=0) and (l2=2) and (v1=1) then  t1:=0;
if  (l1=0) and (l2=2) and (v1=2) then  t1:=1;
if  (l1=1) and (l2=0) and (v1=0) then  t1:=2;
if  (l1=1) and (l2=0) and (v1=1) then  t1:=0;
if  (l1=1) and (l2=0) and (v1=2) then  t1:=1;
if  (l1=1) and (l2=1) and (v1=0) then  t1:=1;
if  (l1=1) and (l2=1) and (v1=1) then  t1:=2;
if  (l1=1) and (l2=1) and (v1=2) then  t1:=0;
if  (l1=1) and (l2=2) and (v1=0) then  t1:=0;
if  (l1=1) and (l2=2) and (v1=1) then  t1:=1;
if  (l1=1) and (l2=2) and (v1=2) then  t1:=2;
```

```
if  (l1=2) and (l2=0) and (v1=0) then  t1:=0;
if  (l1=2) and (l2=0) and (v1=1) then  t1:=1;
if  (l1=2) and (l2=0) and (v1=2) then  t1:=2;
if  (l1=2) and (l2=1) and (v1=0) then  t1:=2;
if  (l1=2) and (l2=1) and (v1=1) then  t1:=0;
if  (l1=2) and (l2=1) and (v1=2) then  t1:=1;
if  (l1=2) and (l2=2) and (v1=0) then  t1:=1;
if  (l1=2) and (l2=2) and (v1=1) then  t1:=2;
if  (l1=2) and (l2=2) and (v1=2) then  t1:=0;
u1:=0;
if  (l1=0) and (l2=0) and (t1=0) then  u1:=1;
if  (l1=0) and (l2=0) and (t1=1) then  u1:=2;
if  (l1=0) and (l2=0) and (t1=2) then  u1:=0;
if  (l1=0) and (l2=1) and (t1=0) then  u1:=0;
if  (l1=0) and (l2=1) and (t1=1) then  u1:=1;
if  (l1=0) and (l2=1) and (t1=2) then  u1:=2;
if  (l1=0) and (l2=2) and (t1=0) then  u1:=2;
if  (l1=0) and (l2=2) and (t1=1) then  u1:=0;
if  (l1=0) and (l2=2) and (t1=2) then  u1:=1;
if  (l1=1) and (l2=0) and (t1=0) then  u1:=2;
if  (l1=1) and (l2=0) and (t1=1) then  u1:=0;
if  (l1=1) and (l2=0) and (t1=2) then  u1:=1;
if  (l1=1) and (l2=1) and (t1=0) then  u1:=1;
if  (l1=1) and (l2=1) and (t1=1) then  u1:=2;
if  (l1=1) and (l2=1) and (t1=2) then  u1:=0;
if  (l1=1) and (l2=2) and (t1=0) then  u1:=0;
if  (l1=1) and (l2=2) and (t1=1) then  u1:=1;
if  (l1=1) and (l2=2) and (t1=2) then  u1:=2;
if  (l1=2) and (l2=0) and (t1=0) then  u1:=0;
if  (l1=2) and (l2=0) and (t1=1) then  u1:=1;
if  (l1=2) and (l2=0) and (t1=2) then  u1:=2;
if  (l1=2) and (l2=1) and (t1=0) then  u1:=2;
if  (l1=2) and (l2=1) and (t1=1) then  u1:=0;
if  (l1=2) and (l2=1) and (t1=2) then  u1:=1;
if  (l1=2) and (l2=2) and (t1=0) then  u1:=1;
if  (l1=2) and (l2=2) and (t1=1) then  u1:=2;
if  (l1=2) and (l2=2) and (t1=2) then  u1:=0;
write ('Enter ',2,' ciphertext element: ');
Readln(v2);
6:if  (v2<>0) and (v2<>1) and (v2<>2) then
begin
Write('Please enter ',2,' ciphertext  correctly: ');
Readln(v2);
goto 6;
end;
if  (l3=0) and (v1=0) and (v2=0) then  u2:=1;
if  (l3=0) and (v1=0) and (v2=1) then  u2:=2;
if  (l3=0) and (v1=0) and (v2=2) then  u2:=0;
if  (l3=0) and (v1=1) and (v2=0) then  u2:=0;
if  (l3=0) and (v1=1) and (v2=1) then  u2:=1;
if  (l3=0) and (v1=1) and (v2=2) then  u2:=2;
if  (l3=0) and (v1=2) and (v2=0) then  u2:=2;
if  (l3=0) and (v1=2) and (v2=1) then  u2:=0;
if  (l3=0) and (v1=2) and (v2=2) then  u2:=1;
if  (l3=1) and (v1=0) and (v2=0) then  u2:=2;
if  (l3=1) and (v1=0) and (v2=1) then  u2:=0;
if  (l3=1) and (v1=0) and (v2=2) then  u2:=1;
if  (l3=1) and (v1=1) and (v2=0) then  u2:=1;
if  (l3=1) and (v1=1) and (v2=1) then  u2:=2;
if  (l3=1) and (v1=1) and (v2=2) then  u2:=0;
if  (l3=1) and (v1=2) and (v2=0) then  u2:=0;
if  (l3=1) and (v1=2) and (v2=1) then  u2:=1;
if  (l3=1) and (v1=2) and (v2=2) then  u2:=2;
if  (l3=2) and (v1=0) and (v2=0) then  u2:=0;
if  (l3=2) and (v1=0) and (v2=1) then  u2:=1;
if  (l3=2) and (v1=0) and (v2=2) then  u2:=2;
if  (l3=2) and (v1=1) and (v2=0) then  u2:=2;
```

```
if  (l3=2) and (v1=1) and (v2=1) then  u2:=0;
if  (l3=2) and (v1=1) and (v2=2) then  u2:=1;
if  (l3=2) and (v1=2) and (v2=0) then  u2:=1;
if  (l3=2) and (v1=2) and (v2=1) then  u2:=2;
if  (l3=2) and (v1=2) and (v2=2) then  u2:=0;
A[1]:=u1;
A[2]:=u2;
B[1]:=v1;
B[2]:=v2;
for k:=3 to n do
begin
Write('Enter ',k,' ciphertext element: ');
Readln(B[k]);
7:if  (B[k]<>0) and (B[k]<>1) and (B[k]<>2) then
begin
Write('Please enter ',k,' ciphertext element correctly: ');
Readln(B[k]);
goto 7;
end;
t3:=0;
if k mod 2=1 then
begin
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=0) then t3:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=1) then t3:=2;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=2) then t3:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=0) then t3:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=1) then t3:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=2) then t3:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=0) then t3:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=1) then t3:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=2) then t3:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=0) then t3:=2;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=1) then t3:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=2) then t3:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=0) then t3:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=1) then t3:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=2) then t3:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=0) then t3:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=1) then t3:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=2) then t3:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=0) then t3:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=1) then t3:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=2) then t3:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=0) then t3:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=1) then t3:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=2) then t3:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=0) then t3:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=1) then t3:=2;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=2) then t3:=0;
if  (B[k-2]=0)and (B[k-1]=0) and (t3=0) then A[k]:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (t3=1) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=0) and (t3=2) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (t3=0) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (t3=1) then A[k]:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (t3=2) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (t3=0) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (t3=1) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (t3=2) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (t3=0) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=0) and (t3=1) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (t3=2) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (t3=0) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (t3=1) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (t3=2) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (t3=0) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (t3=1) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (t3=2) then A[k]:=2;
```

```
if  (B[k-2]=2)and (B[k-1]=0) and (t3=0) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (t3=1) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (t3=2) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (t3=0) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (t3=1) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (t3=2) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (t3=0) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (t3=1) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=2) and (t3=2) then A[k]:=0;
end;
if k mod 2=0 then
begin
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=0) then A[k]:=1;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=1) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=0) and (B[k]=2) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=0)and (B[k-1]=1) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=0) then A[k]:=2;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=1) then A[k]:=0;
if  (B[k-2]=0)and (B[k-1]=2) and (B[k]=2) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=0) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=1) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=0) and (B[k]=2) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=0) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=1) then A[k]:=2;
if  (B[k-2]=1)and (B[k-1]=1) and (B[k]=2) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=1)and (B[k-1]=2) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=0) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=1) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=0) and (B[k]=2) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=0) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=1) then A[k]:=0;
if  (B[k-2]=2)and (B[k-1]=1) and (B[k]=2) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=0) then A[k]:=1;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=1) then A[k]:=2;
if  (B[k-2]=2)and (B[k-1]=2) and (B[k]=2) then A[k]:=0;
end;
end;
begin
    Write('Ciphertext: ');
    for i := 1 to n do write( B[i]);
    writeln;
    Write('Plaintext : ');
    for i := 1 to n do write( A[i]);
   end;
end.
```

Таблица А2.2. Выводы по программам А2.7.-А2.10. (processor: Intel (R) Core (TM) i3-8130 U CPU @ 220GHz)								
Порядок и арность группоида (m и a)	**Длина текста n**	**Используемый алгоритм**	**Количество используемых лидеров (k) и их значения**	**Открытый текст U**	**Зашифрованный текст V**	**Средняя скорость обработки (ms)**		**Оценка сложности алгоритма $O(f(n))$**
						без проверки ввода данных	**с проверкой**	
$m = a = 3$	$n = 9$	Обобщенный Алгоритм 1 для шифрования	$k = 4,$ $l_1 = 1, l_2 = 2,$ $l_3 = 0, l_4 = 1$	021022110	122012212	79.8288	89,0726	линейная $O(n)$
$m = a = 3$	$n = 9$	Обобщенный Алгоритм 1 для дешифрования	$k = 4,$ $l_1 = 1, l_2 = 2,$ $l_3 = 0, l_4 = 1$	021022110	122012212	76.8204	84.2277	линейная $O(n)$
$m = a = 3$	$n = 9$	Обобщенный Алгоритм 2 для шифрования	$k = 3, l_1 = 1,$ $l_2 = 2,\ l_3 = 0$	021022110	011221022	121.2317	129,9153	линейная $O(n)$
$m = a = 3$	$n = 9$	Обобщенный Алгоритм 2 для дешифрования	$k = 3, l_1 = 1,$ $l_2 = 2,\ l_3 = 0$	021022110	011221022	117.7558	124,6386	линейная $O(n)$

Приложение 3. Таблицы процессов, функций шифрования и дешифрования

Таблица А3.1. Значения функции шифрования f для Примера 2.4.7

№	Значение	№	Значение	№	Значение
(1)	$f(0,0,0)=1$	(10)	$f(1,0,0)=2$	(19)	$f(2,0,0)=0$
(2)	$f(0,0,1)=2$	(11)	$f(1,0,1)=0$	(20)	$f(2,0,1)=1$
(3)	$f(0,0,2)=2$	(12)	$f(1,0,2)=0$	(21)	$f(2,0,2)=1$
(4)	$f(0,1,0)=2$	(13)	$f(1,1,0)=0$	(22)	$f(2,1,0)=1$
(5)	$f(0,1,1)=0$	(14)	$f(1,1,1)=1$	(23)	$f(2,1,1)=2$
(6)	$f(0,1,2)=0$	(15)	$f(1,1,2)=1$	(24)	$f(2,1,2)=2$
(7)	$f(0,2,0)=2$	(16)	$f(1,2,0)=0$	(25)	$f(2,2,0)=1$
(8)	$f(0,2,1)=0$	(17)	$f(1,2,1)=1$	(26)	$f(2,2,1)=2$
(9)	$f(0,2,2)=0$	(18)	$f(1,2,2)=1$	(27)	$f(2,2,2)=2$

Таблица А3.2. Значения функции дешифрования $^{(1,4)}f$ для Примера 2.4.7

№	Значение	№	Значение	№	Значение
(1)	$^{(1,4)}f(0,0,0)=2$	(10)	$^{(1,4)}f(1,0,0)=0$	(19)	$^{(1,4)}f(2,0,0)=1$
(2)	$^{(1,4)}f(0,0,1)=1$	(11)	$^{(1,4)}f(1,0,1)=2$	(20)	$^{(1,4)}f(2,0,1)=0$
(3)	$^{(1,4)}f(0,0,2)=1$	(12)	$^{(1,4)}f(1,0,2)=2$	(21)	$^{(1,4)}f(2,0,2)=0$
(4)	$^{(1,4)}f(0,1,0)=1$	(13)	$^{(1,4)}f(1,1,0)=2$	(22)	$^{(1,4)}f(2,1,0)=0$
(5)	$^{(1,4)}f(0,1,1)=0$	(14)	$^{(1,4)}f(1,1,1)=1$	(23)	$^{(1,4)}f(2,1,1)=2$
(6)	$^{(1,4)}f(0,1,2)=0$	(15)	$^{(1,4)}f(1,1,2)=1$	(24)	$^{(1,4)}f(2,1,2)=2$
(7)	$^{(1,4)}f(0,2,0)=1$	(16)	$^{(1,4)}f(1,2,0)=2$	(25)	$^{(1,4)}f(2,2,0)=0$
(8)	$^{(1,4)}f(0,2,1)=0$	(17)	$^{(1,4)}f(1,2,1)=1$	(26)	$^{(1,4)}f(2,2,1)=2$
(9)	$^{(1,4)}f(0,2,2)=0$	(18)	$^{(1,4)}f(1,2,2)=1$	(27)	$^{(1,4)}f(2,2,2)=2$

Таблица А3.3. Значения функции шифрования f для Примера 2.5.4

№	Значение	№	Значение	№	Значение	№	Значение
(1)	$T_{0,0,0}0 = 2$	(21)	$T_{0,2,0}2 = 0$	(41)	$T_{1,1,1}1 = 1$	(61)	$T_{2,0,2}0 = 2$
(2)	$T_{0,0,0}1 = 0$	(22)	$T_{0,2,1}0 = 0$	(42)	$T_{1,1,1}2 = 2$	(62)	$T_{2,0,2}1 = 0$
(3)	$T_{0,0,0}2 = 1$	(23)	$T_{0,2,1}1 = 1$	(43)	$T_{1,1,2}0 = 2$	(63)	$T_{2,0,2}2 = 1$
(4)	$T_{0,0,1}0 = 1$	(24)	$T_{0,2,1}2 = 2$	(44)	$T_{1,1,2}1 = 0$	(64)	$T_{2,1,0}0 = 2$
(5)	$T_{0,0,1}1 = 2$	(25)	$T_{0,2,2}0 = 2$	(45)	$T_{1,1,2}2 = 1$	(65)	$T_{2,1,0}1 = 0$
(6)	$T_{0,0,1}2 = 0$	(26)	$T_{0,2,2}1 = 0$	(46)	$T_{1,2,0}0 = 2$	(66)	$T_{2,1,0}2 = 1$
(7)	$T_{0,0,2}0 = 0$	(27)	$T_{0,2,2}2 = 1$	(47)	$T_{1,2,0}1 = 0$	(67)	$T_{2,1,1}0 = 1$
(8)	$T_{0,0,2}1 = 1$	(28)	$T_{1,0,0}0 = 0$	(48)	$T_{1,2,0}2 = 1$	(68)	$T_{2,1,1}1 = 2$
(9)	$T_{0,0,2}2 = 2$	(29)	$T_{1,0,0}1 = 1$	(49)	$T_{1,2,1}0 = 1$	(69)	$T_{2,1,1}2 = 0$
(10)	$T_{0,1,0}0 = 0$	(30)	$T_{1,0,0}2 = 2$	(50)	$T_{1,2,1}1 = 2$	(70)	$T_{2,1,2}0 = 0$
(11)	$T_{0,1,0}1 = 1$	(31)	$T_{1,0,1}0 = 2$	(51)	$T_{1,2,1}2 = 0$	(71)	$T_{2,1,2}1 = 1$
(12)	$T_{0,1,0}2 = 2$	(32)	$T_{1,0,1}1 = 0$	(52)	$T_{1,2,2}0 = 0$	(72)	$T_{2,1,2}2 = 2$
(13)	$T_{0,1,1}0 = 2$	(33)	$T_{1,0,1}2 = 1$	(53)	$T_{1,2,2}1 = 1$	(73)	$T_{2,2,0}0 = 0$
(14)	$T_{0,1,1}1 = 0$	(34)	$T_{1,0,2}0 = 1$	(54)	$T_{1,2,2}2 = 2$	(74)	$T_{2,2,0}1 = 1$
(15)	$T_{0,1,1}2 = 1$	(35)	$T_{1,0,2}1 = 2$	(55)	$T_{2,0,0}0 = 1$	(75)	$T_{2,2,0}2 = 2$
(16)	$T_{0,1,2}0 = 1$	(36)	$T_{1,0,2}2 = 0$	(56)	$T_{2,0,0}1 = 2$	(76)	$T_{2,2,1}0 = 2$
(17)	$T_{0,1,2}1 = 2$	(37)	$T_{1,1,0}0 = 1$	(57)	$T_{2,0,0}2 = 0$	(77)	$T_{2,2,1}1 = 0$
(18)	$T_{0,1,2}2 = 0$	(38)	$T_{1,1,0}1 = 2$	(58)	$T_{2,0,1}0 = 0$	(78)	$T_{2,2,1}2 = 1$
(19)	$T_{0,2,0}0 = 1$	(39)	$T_{1,1,0}2 = 0$	(59)	$T_{2,0,1}1 = 1$	(79)	$T_{2,2,2}0 = 1$
(20)	$T_{0,2,0}1 = 2$	(40)	$T_{1,1,1}0 = 0$	(60)	$T_{2,0,1}2 = 2$	(80)	$T_{2,2,2}1 = 2$
						(81)	$T_{2,2,2}2 = 0$

Таблица А3.4. Процесс дешифрования для Примера 3.1.4 (Усеченная атака Войводы)

$u_1 = l\backslash q_1 =$ $= 2\backslash 0 = 1$	$u_{11} = q_5\backslash q_1 =$ $= 4\backslash 0 = 3$	$u_{21}=q_4\backslash q_2 =$ $= 3\backslash 1 = 2$	$u_{31}=q_3\backslash q_3 =$ $= 2\backslash 2 = 0$	$u_{41} = q_2\backslash q_4 =$ $= 1\backslash 3 = 0$
$u_2 = q_1\backslash q_1 ==$ $0\backslash 0 = 2$	$u_{12}=q_1\backslash q_6 =$ $= 0\backslash 5 = 5$	$u_{22}=q_2\backslash q_5 =$ $= 1\backslash 4 = 2$	$u_{32}=q_3\backslash q_4 =$ $= 2\backslash 3 = 3$	$u_{42}=q_4\backslash q_3 =$ $= 3\backslash 2 = 4$
$u_3 = q_1\backslash q_1 =$ $= 0\backslash 0 = 2$	$u_{13}=q_6\backslash q_2 =$ $= 5\backslash 1 = 1$	$u_{23}=q_5\backslash q_2 =$ $= 4\backslash 1 = 5$	$u_{33} = q_4\backslash q_3 =$ $= 3\backslash 2 = 4$	$u_{43} = q_3\backslash q_4 =$ $= 2\backslash 3 = 3$
$u_4 = q_1\backslash q_2 =$ $= 0\backslash 1 = 0$	$u_{14}=q_2\backslash q_1 =$ $= 1\backslash 0 = 5$	$u_{24} = q_2\backslash q_6 =$ $= 1\backslash 5 = 4$	$u_{34} = q_3\backslash q_5 =$ $= 2\backslash 4 = 5$	$u_{44} = q_4\backslash q_4 =$ $= 3\backslash 3 = 5$
$u_5 = q_2\backslash q_1 =$ $= 1\backslash 0 = 5$	$u_{15} = q_1\backslash q_2 =$ $= 0\backslash 1 = 0$	$u_{25} = q_6\backslash q_3 =$ $= 5\backslash 2 = 5$	$u_{35} = q_5\backslash q_3 =$ $= 4\backslash 2 = 2$	$u_{45} = q_4\backslash q_4 =$ $= 3\backslash 3 = 5$
$u_7 = q_1\backslash q_3 =$ $= 0\backslash 2 = 3$	$u_{16} = q_2\backslash q_2 =$ $= 1\backslash 1 = 3$	$u_{26} = q_3\backslash q_1 =$ $= 2\backslash 0 = 1$	$u_{36} = q_3\backslash q_6 =$ $= 2\backslash 5 = 2$	$u_{46} = q_4\backslash q_5 =$ $= 3\backslash 4 = 1$
$u_7 = q_3\backslash q_1 =$ $= 2\backslash 0 = 1$	$u_{17} = q_2\backslash q_2 =$ $= 1\backslash 1 = 3$	$u_{27} = q_1\backslash q_3 =$ $= 0\backslash 2 = 3$	$u_{37} = q_6\backslash q_4 =$ $= 5\backslash 3 = 2$	$u_{47} = q_5\backslash q_4 =$ $= 4\backslash 3 = 4$
$u_8 = q_1\backslash q_4 =$ $= 0\backslash 3 = 1$	$u_{18} = q_2\backslash q_3 =$ $= 1\backslash 2 = 1$	$u_{28} = q_3\backslash q_2 =$ $= 2\backslash 1 = 4$	$u_{38} = q_4\backslash q_1 =$ $= 3\backslash 0 = 0$	$u_{48} = q_4\backslash q_6 =$ $= 3\backslash 5 = 3$
$u_9 = q_4\backslash q_1 =$ $= 3\backslash 0 = 0$	$u_{19} = q_3\backslash q_2 =$ $= 2\backslash 1 = 4$	$u_{29} = q_2\backslash q_3 =$ $= 1\backslash 2 = 1$	$u_{39} = q_1\backslash q_4 =$ $= 0\backslash 3 = 1$	$u_{49} = q_6\backslash q_5 =$ $= 5\backslash 4 = 3$
$u_{10} = q_1\backslash q_5 =$ $= 0\backslash 4 = 4$	$u_{20} = q_2\backslash q_4 =$ $= 1\backslash 3 = 0$	$u_{30} = q_3\backslash q_3 =$ $= 2\backslash 2 = 2$	$u_{40} = q_4\backslash q_2 =$ $= 3\backslash 1 = 2$	

Таблица А3.5. Процесс дешифрования для Примера 3.1.4.

(модифицированная атака)

$u_1 = l\backslash q_1 =$ $= 2\backslash 0 = 1$	$u_6 = q_3\backslash q_3 =$ $= 2\backslash 2 = 0$	$u_{11}=q_5\backslash q_6 =$ $= 4\backslash 5 = 1$	$u_{16}=q_3\backslash q_2 =$ $= 2\backslash 1 = 4$	$u_{21} = q_4\backslash q_6 =$ $= 3\backslash 5 = 3$
$u_2 \backslash q_1 == 0\backslash$ $0 = 2$	$u_7= q_3\backslash q_4 =$ $= 2\backslash 3 = 3$	$u_{12}=q_6\backslash q_6 =$ $= 5\backslash 5 = 0$	$u_{17}=q_2\backslash q_4 =$ $= 1\backslash 3 = 0$	$u_{22}=q_6\backslash q_5 =$ $= 5\backslash 4 = 3$
$u_3 = q_1\backslash q_2 =$ $= 0\backslash 1 = 0$	$u_8= q_4\backslash q_4 =$ $= 3\backslash 3 = 5$	$u_{13}=q_6\backslash q_2 =$ $= 5\backslash 1 = 1$	$u_{18} = q_4\backslash q_3 =$ $= 3\backslash 2 = 4$	$u_{23} = q_5\backslash q_1 =$ $= 4\backslash 0 = 3$
$u_4 = q_2\backslash q_2 =$ $= 1\backslash 1 = 3$	$u_9=q_4\backslash q_5 =$ $= 3\backslash 4 = 1$	$u_{14} = q_2\backslash q_1 =$ $= 1\backslash 0 = 5$	$u_{19} = q_3\backslash q_5 =$ $= 2\backslash 4 = 5$	$u_{24} = q_1\backslash q_6 =$ $= 0\backslash 5 = 5$
$u_5 = q_2\backslash q_3 =$ $= 1\backslash 2 = 1$	$u_{10} = q_5\backslash q_5 =$ $= 4\backslash 4 = 0$	$u_{15} = q_1\backslash q_3 =$ $= 0\backslash 2 = 3$	$u_{20} = q_5\backslash q_4 =$ $= 4\backslash 3 = 4$	$u_{25} = q_6\backslash q_3 =$ $= 5\backslash 2 = 5$
				$u_{26} = q_3\backslash q_1 =$ $= 2\backslash 0 = 1$

Таблица А3.6. Процесс дешифрования для Примера 3.3.1.

$u_1 = l\backslash q_1 =$ $= 3\backslash 0 = 0$	$u_{11} = q_5\backslash q_2 =$ $= 4\backslash 1 = 1$	$u_{21}=q_5\backslash q_3 =$ $= 4\backslash 2 = 0$	$u_{31}=q_5\backslash q_4 =$ $= 4\backslash 3 = 4$	$u_{41} = q_5\backslash q_5 =$ $= 4\backslash 4 = 3$
$u_2 = q_1\backslash q_1 =$ $= 0\backslash 0 = 4$	$u_{12}=q_2\backslash q_1 =$ $= 1\backslash 0 = 3$	$u_{22}=q_3\backslash q_1 =$ $= 2\backslash 0 = 4$	$u_{32}=q_4\backslash q_1 =$ $= 3\backslash 0 = 0$	$u_{42}=q_5\backslash q_1 =$ $= 4\backslash 0 = 2$
$u_3 = q_1\backslash q_1 =$ $= 0\backslash 0 = 4$	$u_{13}=q_1\backslash q_2 =$ $= 0\backslash 1 = 3$	$u_{23}=q_1\backslash q_3 =$ $= 0\backslash 2 = 2$	$u_{33} = q_1\backslash q_4 =$ $= 0\backslash 3 = 1$	$u_{43} = q_1\backslash q_5 =$ $= 0\backslash 4 = 0$
$u_4 = q_1\backslash q_2 =$ $= 0\backslash 1 = 3$	$u_{14}=q_2\backslash q_2 =$ $= 1\backslash 1 = 2$	$u_{24} = q_3\backslash q_2 =$ $= 2\backslash 1 = 0$	$u_{34} = q_4\backslash q_2 =$ $= 3\backslash 1 = 2$	$u_{44} = q_5\backslash q_2 =$ $= 4\backslash 1 = 1$
$u_5 = q_2\backslash q_1 =$ $= 1\backslash 0 = 3$	$u_{15} = q_2\backslash q_2 =$ $= 1\backslash 1 = 2$	$u_{25} = q_2\backslash q_3 =$ $= 1\backslash 2 = 1$	$u_{35} = q_2\backslash q_4 =$ $= 1\backslash 3 = 0$	$u_{45} = q_2\backslash q_5 =$ $= 1\backslash 4 = 4$
$u_7 = q_1\backslash q_3 =$ $= 0\backslash 2 = 2$	$u_{16} = q_2\backslash q_3 =$ $= 1\backslash 2 = 1$	$u_{26} = q_3\backslash q_3 =$ $= 2\backslash 2 = 3$	$u_{36} = q_4\backslash q_3 =$ $= 3\backslash 2 = 1$	$u_{46} = q_5\backslash q_3 =$ $= 4\backslash 2 = 0$
$u_7 = q_3\backslash q_1 =$ $= 2\backslash 0 = 4$	$u_{17} = q_3\backslash q_2 =$ $= 2\backslash 1 = 0$	$u_{27} = q_3\backslash q_3 =$ $= 2\backslash 2 = 3$	$u_{37} = q_3\backslash q_4 =$ $= 2\backslash 3 = 1$	$u_{47} = q_3\backslash q_5 =$ $= 2\backslash 4 = 2$
$u_8 = q_1\backslash q_4 =$ $= 0\backslash 3 = 1$	$u_{18} = q_2\backslash q_4 =$ $= 1\backslash 3 = 0$	$u_{28} = q_3\backslash q_4 =$ $= 2\backslash 3 = 1$	$u_{38} = q_4\backslash q_4 =$ $= 3\backslash 3 = 3$	$u_{48} = q_5\backslash q_4 =$ $= 4\backslash 3 = 4$
$u_9 = q_4\backslash q_1 =$ $= 3\backslash 0 = 0$	$u_{19} = q_4\backslash q_2 =$ $= 3\backslash 1 = 2$	$u_{29} = q_4\backslash q_3 =$ $= 3\backslash 2 = 1$	$u_{39} = q_4\backslash q_4 =$ $= 3\backslash 3 = 3$	$u_{49} = q_4\backslash q_5 =$ $= 3\backslash 4 = 4$
$u_{10} = q_1\backslash q_5 =$ $= 0\backslash 4 = 0$	$u_{20} = q_2\backslash q_5 =$ $= 1\backslash 4 = 4$	$u_{30} = q_3\backslash q_5 =$ $= 2\backslash 4 = 2$	$u_{40} = q_4\backslash q_5 =$ $= 3\backslash 4 = 4$	$u_{50} = q_5\backslash q_5 =$ $= 4\backslash 4 = 3$

Таблица А3.7. Процесс шифрования для Примера 3.3.3.

(Атака Войводы с дискретным вводом символов)

$v_1 = l * q_1 = 3 * 0 = 0,$ $v_2 = (l * q_1) * q_1 = 0 * 0 = 4$	04	$v_{31} = l * q_4 = 3 * 3 = 3,\ v_{32} = (l *$ $q_4) * q_1 = 3 * 0 = 0$	30
$v_3 = l * q_1 = 3 * 0 = 0,$ $v_4 = (l * q_1) * q_2 = 0 * 1 = 3$	03	$v_{33} = l * q_4 = 3 * 3 = 3,\ v_{34} = (l *$ $q_4) * q_2 = 3 * 1 = 2$	32
$v_5 = l * q_1 = 3 * 0 = 0,$ $v_6 = (l * q_1) * q_3 = 0 * 2 = 2$	02	$v_{35} = l * q_4 = 3 * 3 = 3,\ v_{36} = (l *$ $q_4) * q_3 = 3 * 2 = 1$	31
$v_7 = l * q_1 = 3 * 0 = 0,$ $v_8 = (l * q_1) * q_4 = 0 * 3 = 1$	01	$v_{37} = l * q_4 = 3 * 3 = 3,\ v_{38} = (l *$ $q_4) * q_4 = 3 * 3 = 3$	33
$v_9 = l * q_1 = 3 * 0 = 0,$ $v_{10} = (l * q_1) * q_5 = 0 * 4 = 0$	00 (лишний)	$v_{39} = l * q_4 = 3 * 3 = 3,\ v_{40} = (l *$ $q_4) * q_5 = 3 * 4 = 4$	34 (лишний)

$v_{11} = l * q_2 = 3 * 1 = 2,\ v_{12} = (l * q_2) * q_1 = 2 * 0 = 1$	21	$v_{41} = l * q_5 = 3 * 4 = 4,\ v_{42} = (l * q_5) * q_1 = 4 * 0 = 2$	42
$v_{13} = l * q_2 = 3 * 1 = 2,\ v_{14} = (l * q_2) * q_2 = 2 * 1 = 3$	23	$v_{43} = l * q_5 = 3 * 4 = 4,\quad v_{44} = (l * q_5) * q_2 = 4 * 1 = 1$	41
$v_{15} = l * q_2 = 3 * 1 = 2,\ v_{16} = (l * q_2) * q_3 = 2 * 2 = 4$	24	$v_{45} = l * q_5 = 3 * 4 = 4,\ v_{46} = (l * q_5) * q_3 = 4 * 2 = 0$	40
$v_{17} = l * q_2 = 3 * 1 = 2,\ v_{18} = (l * q_2) * q_4 = 2 * 3 = 2$	22	$v_{47} = l * q_5 = 3 * 4 = 4,\ v_{48} = (l * q_5) * q_4 = 4 * 3 = 4$	44
$v_{19} = l * q_2 = 3 * 1 = 2,$ $v_{20} = (l * q_2) * q_5 = 2 * 4 = 0$	20 (лишний)	$v_{49} = l * q_5 = 3 * 4 = 4,\ v_{50} = (l * q_5) * q_5 = 4 * 4 = 3$	43 (лишний)
$v_{21} = l * q_3 = 3 * 2 = 1,\ v_{22} = (l * q_3) * q_1 = 1 * 0 = 3$	13		
$v_{23} = l * q_3 = 3 * 2 = 1,\ v_{24} = (l * q_3) * q_2 = 1 * 1 = 2$	12		
$v_{25} = l * q_3 = 3 * 2 = 1,\ \ v_{26} = (l * q_3) * q_3 = 1 * 2 = 1$	11		
$v_{27} = l * q_3 = 3 * 2 = 1,$ $v_{28} = (l * q_3) * q_4 = 1 * 3 = 0$	10		
$v_{29} = l * q_3 = 3 * 2 = 1,$ $v_{30} = (l * q_3) * q_5 = 1 * 4 = 4$	14 (лишний)		

Таблица А3.8. Процесс шифрования для Примера 3.3.3

(поточная атака текстом Войводы)

$v_1 = l * q_1 = 3 * 0 = 0$	0	$v_{11} = v_{10} * q_2 = 3 * 1 = 2$	2	$v_{21} = v_{20} * q_3 = 4 * 2 = 0$	0
$v_2 = v_1 * q_1 = 0 * 0 = 4$	4	$v_{12} = v_{11} * q_1 = 2 * 0 = 1$	1	$v_{22} = v_{21} * q_1 = 0 * 0 = 4$	4
$v_3 = v_2 * q_1 = 4 * 0 = 2$	2	$v_{13} = v_{12} * q_2 = 1 * 1 = 2$	2	$v_{23} = v_{22} * q_3 = 4 * 2 = 0$	0
$v_4 = v_3 * q_2 = 2 * 1 = 3$	3	$v_{14} = v_{13} * q_2 = 2 * 1 = 3$	3	$v_{24} = v_{23} * q_2 = 0 * 1 = 3$	3
$v_5 = v_4 * q_1 = 3 * 0 = 0$	0	$v_{15} = v_{14} * q_2 = 3 * 1 = 2$	2	$v_{25} = v_{24} * q_3 = 3 * 2 = 1$	1
$v_6 = v_5 * q_3 = 0 * 2 = 2$	2	$v_{16} = v_{15} * q_3 = 2 * 2 = 4$	4	$v_{26} = v_{25} * q_3 = 1 * 2 = 1$	1
$v_7 = v_6 * q_1 = 2 * 0 = 1$	1	$v_{17} = v_{16} * q_2 = 4 * 1 = 1$	1	$v_{27} = v_{26} * q_3 = 1 * 2 = 1$	1
$v_8 = v_7 * q_4 = 1 * 3 = 0$	0	$v_{18} = v_{17} * q_4 = 1 * 3 = 0$	0	$v_{28} = v_{27} * q_4 = 1 * 3 = 0$	0
$v_9 = v_8 * q_1 = 0 * 0 = 4$	4	$v_{19} = v_{18} * q_2 = 0 * 1 = 3$	3	$v_{29} = v_{28} * q_3 = 0 * 2 = 2$	2
$v_{10} = v_9 * q_5 = 4 * 4 = 3$	3	$v_{20} = v_{19} * q_5 = 3 * 4 = 4$	4	$v_{30} = v_{29} * q_5 = 2 * 4 = 0$	0
				$v_{31} = v_{30} * q_4 = 0 * 3 = 1$	1
				$v_{32} = v_{31} * q_1 = 1 * 0 = 3$	3

Таблица А3.9. Процесс шифрования для Примера 3.3.3

(поточная атака минимальным текстом)

$v_1 = l * q_1 = 3 * 0 = 0$	0	$v_8 = v_7 * q_4 = 1 * 3 = 0$	0	$v_{15} = v_{14} * q_1 = 2 * 0 = 1$	1
$v_2 = v_1 * q_1 = 0 * 0 = 4$	4	$v_9 = v_8 * q_2 = 0 * 1 = 3$	3	$v_{16} = v_{15} * q_1 = 1 * 0 = 3$	3
$v_3 = v_2 * q_2 = 4 * 1 = 1$	1	$v_{10} = v_9 * q_5 = 3 * 4 = 4$	4	$v_{17} = v_{16} * q_3 = 3 * 2 = 1$	1
$v_4 = v_3 * q_2 = 1 * 1 = 2$	2	$v_{11} = v_{10} * q_5 = 4 * 4 = 3$	3	$v_{18} = v_{17} * q_5 = 1 * 4 = 4$	4
$v_5 = v_4 * q_3 = 2 * 2 = 4$	4	$v_{12} = v_{11} * q_4 = 3 * 3 = 3$	3	$v_{19} = v_{18} * q_1 = 4 * 0 = 2$	2
$v_6 = v_5 * q_3 = 4 * 2 = 0$	0	$v_{13} = v_{12} * q_2 = 3 * 1 = 2$	2	$v_{20} = v_{19} * q_5 = 2 * 4 = 0$	0
$v_7 = v_6 * q_4 = 0 * 3 = 1$	1	$v_{14} = v_{13} * q_4 = 2 * 3 = 2$	2	$v_{21} = v_{20} * q_3 = 0 * 2 = 2$	2

Таблица А3.10. Процесс шифрования для Примера 3.3.3

(модифицированная атака)

$v_1 = l * q_1 = 3 * 0 = 0, v_2 = (l * q_1) * q_1 = 0 * 0 = 4$	04
$v_3 = l * q_2 = 3 * 1 = 2,\ v_4 = (l * q_2) * q_2 = 2 * 1 = 3$	23
$v_5 = l * q_3 = 3 * 2 = 1,\ v_6 = (l * q_3) * q_3 = 1 * 2 = 1$	11
$v_7 = l * q_4 = 3 * 3 = 3,\ v_8 = (l * q_4) * q_4 = 3 * 3 = 3$	33
$v_9 = l * q_5 = 3 * 4 = 4,\ v_{10} = (l * q_5) * q_5 = 4 * 4 = 3$	43
$v_{11} = l * q_2 = 3 * 1 = 2,\ v_{12} = (l * q_2) * q_1 = 2 * 0 = 1$	21
$v_{13} = l * q_3 = 3 * 2 = 1,\ v_{14} = (l * q_3) * q_2 = 1 * 1 = 2$	12
$v_{15} = l * q_4 = 3 * 3 = 3,\ v_{16} = (l * q_4) * q_3 = 3 * 2 = 1$	31
$v_{17} = l * q_5 = 3 * 4 = 4,\ v_{18} = (l * q_5) * q_4 = 4 * 3 = 4$	44
$v_{19} = l * q_1 = 3 * 0 = 0,\ v_{20} = (l * q_1) * q_5 = 0 * 4 = 0$	00
$v_{21} = l * q_3 = 3 * 2 = 1,\ v_{22} = (l * q_3) * q_1 = 1 * 0 = 3$	13
$v_{23} = l * q_4 = 3 * 3 = 3,\ v_{23} = (l * q_4) * q_2 = 3 * 1 = 2$	32
$v_{25} = l * q_5 = 3 * 4 = 4,\ v_{26} = (l * q_5) * q_3 = 4 * 2 = 0$	40
$v_{27} = l * q_1 = 3 * 0 = 0,\ v_{28} = (l * q_1) * q_4 = 0 * 3 = 1$	01
$v_{29} = l * q_2 = 3 * 1 = 2,\ v_{30} = (l * q_2) * q_5 = 2 * 4 = 0$	20
$v_{31} = l * q_4 = 3 * 3 = 3,\ v_{32} = (l * q_4) * q_1 = 3 * 0 = 0$	30
$v_{33} = l * q_5 = 3 * 4 = 4,\ v_{34} = (l * q_5) * q_2 = 4 * 1 = 1$	41
$v_{35} = l * q_1 = 3 * 0 = 0,\ v_{36} = (l * q_1) * q_3 = 0 * 2 = 2$	02
$v_{37} = l * q_2 = 3 * 1 = 2,\ v_{38} = (l * q_2) * q_4 = 2 * 3 = 2$	22
$v_{39} = l * q_3 = 3 * 2 = 1,\ v_{40} = (l * q_3) * q_5 = 1 * 4 = 4$	14

Таблица А3.11. Процесс дешифрования для Примера 3.4.1 **(модифицированная атака)**

$u_1 = q_1/l = 0/2 = 0$	$u_{12} = q_1/q_2 = 0/1 = 2$
$u_2 = q_1/q_1 = 0/0 = 1$	$u_{13} = q_3/q_1 = 2/0 = 2$
$u_3 = q_2/q_1 = 1/0 = 3$	$u_{14} = q_2/q_3 = 1/2 = 2$
$u_4 = q_2/q_2 = 1/1 = 4$	$u_{15} = q_4/q_2 = 3/1 = 3$
$u_5 = q_3/q_2 = 2/1 = 1$	$u_{16} = q_3/q_4 = 2/3 = 0$
$u_6 = q_3/q_3 = 2/2 = 3$	$u_{17} = q_5/q_3 = 4/2 = 1$
$u_7 = q_4/q_3 = 3/2 = 4$	$u_{18} = q_4/q_5 = 3/4 = 1$
$u_8 = q_4/q_4 = 3/3 = 2$	$u_{19} = q_1/q_4 = 0/3 = 3$
$u_9 = q_5/q_4 = 4/3 = 4$	$u_{20} = q_5/q_1 = 4/0 = 0$
$u_{10} = q_5/q_5 = 4/4 = 3$	$u_{21} = q_3/q_5 = 2/4 = 2$
$u_{11} = q_2/q_5 = 1/4 = 0$	

Таблица А3.12. Процесс шифрования для Примера 3.4.2 (поточная атака текстом Войводы)

$v_1 = q_1 * l = 0 * 2 = 0$	0	$v_{24} = q_2 * v_{23} = 1 * 2 = 4$	4
$v_2 = q_1 * v_1 = 0 * 0 = 4$	4	$v_{25} = q_3 * v_{24} = 2 * 4 = 2$	2
$v_3 = q_1 * v_2 = 0 * 4 = 1$	1	$v_{26} = q_3 * v_{25} = 2 * 2 = 1$	1
$v_4 = q_2 * v_3 = 1 * 1 = 2$	2	$v_{27} = q_3 * v_{26} = 2 * 1 = 0$	0
$v_5 = q_1 * v_4 = 0 * 2 = 0$	0	$v_{28} = q_4 * v_{27} = 3 * 0 = 1$	1
$v_6 = q_3 * v_5 = 2 * 0 = 2$	2	$v_{29} = q_3 * v_{28} = 2 * 1 = 0$	0
$v_7 = q_1 * v_6 = 0 * 2 = 0$	0	$v_{30} = q_5 * v_{29} = 4 * 0 = 3$	3
$v_8 = q_4 * v_7 = 3 * 0 = 1$	1	$v_{31} = q_4 * v_{30} = 3 * 3 = 0$	0
$v_9 = q_1 * v_8 = 0 * 1 = 4$	4	$v_{32} = q_1 * v_{31} = 0 * 0 = 4$	4
$v_{10} = q_5 * v_9 = 4 * 4 = 0$	0	$v_{33} = q_4 * v_{32} = 3 * 4 = 4$	4
$v_{11} = q_2 * v_{10} = 1 * 0 = 0$	0	$v_{34} = q_2 * v_{33} = 1 * 4 = 3$	3
$v_{12} = q_1 * v_{11} = 0 * 0 = 4$	4	$v_{35} = q_4 * v_{34} = 3 * 3 = 0$	0
$v_{13} = q_2 * v_{12} = 1 * 4 = 3$	3	$v_{36} = q_3 * v_{35} = 2 * 0 = 2$	2
$v_{14} = q_2 * v_{13} = 1 * 3 = 1$	1	$v_{37} = q_4 * v_{36} = 3 * 2 = 2$	2
$v_{15} = q_2 * v_{14} = 1 * 1 = 2$	2	$v_{38} = q_4 * v_{37} = 3 * 2 = 2$	2
$v_{16} = q_3 * v_{15} = 2 * 2 = 1$	1	$v_{39} = q_4 * v_{38} = 3 * 2 = 2$	2
$v_{17} = q_2 * v_{16} = 1 * 1 = 2$	2	$v_{40} = q_5 * v_{39} = 4 * 2 = 3$	3
$v_{18} = q_4 * v_{17} = 3 * 2 = 2$	2	$v_{41} = q_5 * v_{40} = 4 * 3 = 4$	4
$v_{19} = q_2 * v_{18} = 1 * 2 = 4$	4	$v_{42} = q_1 * v_{41} = 0 * 4 = 1$	1
$v_{20} = q_5 * v_{19} = 4 * 4 = 0$	0	$v_{43} = q_5 * v_{42} = 4 * 1 = 1$	1
$v_{21} = q_3 * v_{20} = 2 * 0 = 2$	2	$v_{44} = q_2 * v_{43} = 1 * 1 = 2$	2
$v_{22} = q_1 * v_{21} = 0 * 2 = 0$	0	$v_{45} = q_5 * v_{44} = 4 * 2 = 3$	3
$v_{23} = q_3 * v_{22} = 2 * 0 = 2$	2	$v_{46} = q_3 * v_{45} = 2 * 3 = 3$	3

Таблица А3.13. Процесс шифрования для Примера 3.4.2.

(поточная атака минимальным текстом)

$v_1 = q_1 * l = 0 * 2 = 0$	0	$v_{11} = q_2 * v_{10} = 1 * 1 = 2$	2
$v_2 = q_1 * v_1 = 0 * 0 = 4$	4	$v_{12} = q_2 * v_{11} = 1 * 2 = 4$	4
$v_3 = q_2 * v_2 = 1 * 4 = 3$	3	$v_{13} = q_4 * v_{12} = 3 * 4 = 4$	4
$v_4 = q_2 * v_3 = 1 * 3 = 1$	1	$v_{14} = q_1 * v_{13} = 0 * 4 = 1$	1
$v_5 = q_3 * v_4 = 2 * 1 = 0$	0	$v_{15} = q_4 * v_{14} = 3 * 1 = 3$	3
$v_6 = q_3 * v_5 = 2 * 0 = 2$	2	$v_{16} = q_1 * v_{15} = 0 * 3 = 2$	2
$v_7 = q_4 * v_6 = 3 * 2 = 2$	2	$v_{17} = q_3 * v_{16} = 2 * 2 = 1$	1
$v_8 = q_5 * v_7 = 4 * 2 = 3$	3	$v_{18} = q_1 * v_{17} = 0 * 1 = 4$	4
$v_9 = q_4 * v_8 = 3 * 3 = 0$	0	$v_{19} = q_5 * v_{18} = 4 * 4 = 0$	0
$v_{10} = q_4 * v_9 = 3 * 0 = 1$	1	$v_{20} = q_5 * v_{19} = 4 * 0 = 3$	3
		$v_{21} = q_5 * v_{20} = 4 * 3 = 4$	4

Таблица А3.14. Процесс шифрования для Примера 3.4.2

(модифицированная атака)

$v_1 = 0 * 2 = 0, v_2 = 0 * 0 = 4$	04	$v_{21} = 2 * 2 = 1, v_{22} = 0 * 1 = 4$	14
$v_3 = 1 * 2 = 4, v_4 = 1 * 4 = 3$	43	$v_{23} = 3 * 2 = 2, v_{24} = 1 * 2 = 4$	24
$v_5 = 2 * 2 = 1, v_6 = 2 * 1 = 0$	10	$v_{25} = 4 * 2 = 3, v_{26} = 2 * 3 = 3$	33
$v_7 = 3 * 2 = 2, v_8 = 3 * 2 = 2$	22	$v_{27} = 0 * 2 = 0, v_{28} = 3 * 0 = 1$	01
$v_9 = 4 * 2 = 3, v_{10} = 4 * 3 = 4$	34	$v_{29} = 1 * 2 = 4, v_{30} = 4 * 4 = 0$	40
$v_{11} = 1 * 2 = 4, v_{12} = 0 * 4 = 1$	41	$v_{31} = 3 * 2 = 2, v_{32} = 0 * 2 = 0$	20
$v_{13} = 2 * 2 = 1, v_{14} = 1 * 1 = 2$	12	$v_{33} = 4 * 2 = 3, v_{34} = 1 * 3 = 1$	31
$v_{15} = 3 * 2 = 2, v_{16} = 2 * 2 = 1$	21	$v_{35} = 0 * 2 = 0, v_{36} = 2 * 0 = 2$	02
$v_{17} = 4 * 2 = 3, v_{18} = 3 * 3 = 0$	30	$v_{37} = 1 * 2 = 4, v_{38} = 3 * 4 = 4$	44
$v_{19} = 0 * 2 = 0, v_{20} = 4 * 0 = 3$	03	$v_{39} = 2 * 2 = 1, v_{40} = 4 * 1 = 1$	11

Таблица А3.15. Процесс дешифрования для Примера 4.1.2

$u_1 = {}^{(4,5)}f(l_1, l_2, l_3, v_1) = {}^{(4,5)}f(1,0,0,0) = 2$	$u_{110} = {}^{(4,5)}f(2,2,1,0) = 0$
$u_2 = {}^{(4,5)}f(l_4, l_5, l_6, v_2) = {}^{(4,5)}f(2,1,1,0) = 1$	$u_{111} = {}^{(4,5)}f(2,1,0,0) = 0$
$u_3 = {}^{(4,5)}f(l_7, l_8, l_9, v_3) = {}^{(4,5)}f(0,0,0,0) = 1$	$u_{112} = {}^{(4,5)}f(1,0,0,0) = 2$
$u_4 = {}^{(4,5)}f(v_1, v_2, v_3, v_4) = {}^{(4,5)}f(0,0,0,0) = 1 -$**(1)**	$u_{113} = {}^{(4,5)}f(0,0,0,1) = 2$
$u_5 = {}^{(4,5)}f(0,0,0,0) = 1$	$u_{114} = {}^{(4,5)}f(0,0,1,0) = 2$
$u_6 = {}^{(4,5)}f(0,0,0,0) = 1$	$u_{115} = {}^{(4,5)}f(0,1,0,0) = 1$
$u_7 = {}^{(4,5)}f(0,0,0,0) = 1$	$u_{116} = {}^{(4,5)}f(1,0,0,1) = 0$
$u_8 = {}^{(4,5)}f(0,0,0,1) = 2 -$**(2)**	$u_{117} = {}^{(4,5)}f(0,0,1,1) = 0$
$u_9 = {}^{(4,5)}f(0,0,1,0) = 2 -$**(4)**	$u_{118} = {}^{(4,5)}f(0,1,1,0) = 2$
$u_{10} = {}^{(4,5)}f(0,1,0,0) = 1 -$**(10)**	$u_{119} = {}^{(4,5)}f(1,1,0,0) = 2$
$u_{11} = {}^{(4,5)}f(1,0,0,0) = 2 -$**(28)**	$u_{120} = {}^{(4,5)}f(1,0,0,2) = 1$
$u_{12} = {}^{(4,5)}f(0,0,0,2) = 0 -$**(3)**	$u_{121} = {}^{(4,5)}f(0,0,2,1) = 0$
$u_{13} = {}^{(4,5)}f(0,0,2,0) = 2 -$**(7)**	$u_{122} = {}^{(4,5)}f(0,2,1,0) = 1$
$u_{14} = {}^{(4,5)}f(0,2,0,0) = 0 -$**(19)**	$u_{123} = {}^{(4,5)}f(2,1,0,1) = 1$
$u_{15} = {}^{(4,5)}f(2,0,0,1) = 1 -$**(56)**	$u_{124} = {}^{(4,5)}f(1,0,1,0) = 0$
$u_{16} = {}^{(4,5)}f(0,0,1,0) = 2$	$u_{125} = {}^{(4,5)}f(0,1,0,1) = 2$
$u_{17} = {}^{(4,5)}f(0,1,0,0) = 1$	$u_{126} = {}^{(4,5)}f(1,0,1,0) = 0$
$u_{18} = {}^{(4,5)}f(1,0,0,0) = 2$	$u_{127} = {}^{(4,5)}f(0,1,0,1) = 2$
$u_{19} = {}^{(4,5)}f(0,0,0,1) = 2$	$u_{128} = {}^{(4,5)}f(1,0,1,1) = 1$
$u_{20} = {}^{(4,5)}f(0,0,1,1) = 0 -$**(5)**	$u_{129} = {}^{(4,5)}f(0,1,1,1) = 0$
$u_{21} = {}^{(4,5)}f(0,1,1,0) = 2 -$**(13)**	$u_{130} = {}^{(4,5)}f(1,1,1,0) = 0$
$u_{22} = {}^{(4,5)}f(1,1,0,0) = 2 -$**(37)**	$u_{131} = {}^{(4,5)}f(1,1,0,1) = 0$
$u_{23} = {}^{(4,5)}f(1,0,0,1) = 0 -$**(29)**	$u_{132} = {}^{(4,5)}f(1,0,1,2) = 2$
$u_{24} = {}^{(4,5)}f(0,0,1,2) = 1 -$**(6)**	$u_{133} = {}^{(4,5)}f(0,1,2,1) = 0$
$u_{25} = {}^{(4,5)}f(0,1,2,0) = 2 -$**(16)**	$u_{134} = {}^{(4,5)}f(1,2,1,0) = 2$
$u_{26} = {}^{(4,5)}f(1,2,0,0) = 1 -$**(46)**	$u_{135} = {}^{(4,5)}f(2,1,0,2) = 2$
$u_{27} = {}^{(4,5)}f(2,0,0,2) = 2 -$**(57)**	$u_{136} = {}^{(4,5)}f(1,0,2,0) = 0$
$u_{28} = {}^{(4,5)}f(0,0,2,0) = 2$	$u_{137} = {}^{(4,5)}f(0,2,0,1) = 1$
$u_{29} = {}^{(4,5)}f(0,2,0,0) = 0$	$u_{138} = {}^{(4,5)}f(2,0,1,0) = 1$
$u_{30} = {}^{(4,5)}f(2,0,0,0) = 0 -$**(55)**	$u_{139} = {}^{(4,5)}f(0,1,0,2) = 0$
$u_{31} = {}^{(4,5)}f(0,0,0,2) = 0$	$u_{140} = {}^{(4,5)}f(1,0,2,1) = 1$
$u_{32} = {}^{(4,5)}f(0,0,2,1) = 0 -$**(8)**	$u_{141} = {}^{(4,5)}f(0,2,1,1) = 2$
$u_{33} = {}^{(4,5)}f(0,2,1,0) = 1 -$**(22)**	$u_{142} = {}^{(4,5)}f(2,1,1,0) = 1$
$u_{34} = {}^{(4,5)}f(2,1,0,0) = 0 -$**(64)**	$u_{143} = {}^{(4,5)}f(1,1,0,2) = 1$
$u_{35} = {}^{(4,5)}f(1,0,0,2) = 1 -$**(30)**	$u_{144} = {}^{(4,5)}f(1,0,2,2) = 2$
$u_{36} = {}^{(4,5)}f(0,0,2,2) = 1 -$**(9)**	$u_{145} = {}^{(4,5)}f(0,2,2,1) = 2$
$u_{37} = {}^{(4,5)}f(0,2,2,0) = 1 -$**(25)**	$u_{146} = {}^{(4,5)}f(2,2,1,1) = 1 -$**(77)**
$u_{38} = {}^{(4,5)}f(2,2,0,1) = 0 -$**(74)**	$u_{147} = {}^{(4,5)}f(2,1,1,0) = 1$

$u_{39} = {}^{(4,5)}f(2,0,1,0) = 1$ **−(58)**	$u_{148} = {}^{(4,5)}f(1,1,0,0) = 2$
$u_{40} = {}^{(4,5)}f(0,1,0,0) = 1$	$u_{149} = {}^{(4,5)}f(1,0,0,1) = 0$
$u_{41} = {}^{(4,5)}f(1,0,0,0) = 2$	$u_{150} = {}^{(4,5)}f(0,0,1,1) = 0$
$u_{42} = {}^{(4,5)}f(0,0,0,1) = 2$	$u_{151} = {}^{(4,5)}f(0,1,1,0) = 2$
$u_{43} = {}^{(4,5)}f(0,0,1,0) = 2$	$u_{152} = {}^{(4,5)}f(1,1,0,1) = 0$
$u_{44} = {}^{(4,5)}f(0,1,0,1) = 2$ **−(11)**	$u_{153} = {}^{(4,5)}f(1,0,1,1) = 1$
$u_{45} = {}^{(4,5)}f(1,0,1,0) = 0$ **−(31)**	$u_{154} = {}^{(4,5)}f(0,1,1,1) = 0$
$u_{46} = {}^{(4,5)}f(0,1,0,1) = 2$	$u_{155} = {}^{(4,5)}f(1,1,1,0) = 0$
$u_{47} = {}^{(4,5)}f(1,0,1,0) = 0$	$u_{156} = {}^{(4,5)}f(1,1,0,2) = 1$
$u_{48} = {}^{(4,5)}f(0,1,0,2) = 0$ **−(12)**	$u_{157} = {}^{(4,5)}f(1,0,2,1) = 1$
$u_{49} = {}^{(4,5)}f(1,0,2,0) = 0$ **−(34)**	$u_{158} = {}^{(4,5)}f(0,2,1,1) = 2$
$u_{50} = {}^{(4,5)}f(0,2,0,1) = 1$ **−(20)**	$u_{159} = {}^{(4,5)}f(2,1,1,1) = 2$ **−(68)**
$u_{51} = {}^{(4,5)}f(2,0,1,1) = 2$ **−(59)**	$u_{160} = {}^{(4,5)}f(1,1,1,0) = 0$
$u_{52} = {}^{(4,5)}f(0,1,1,0) = 2$	$u_{161} = {}^{(4,5)}f(1,1,0,1) = 0$
$u_{53} = {}^{(4,5)}f(1,1,0,0) = 2$	$u_{162} = {}^{(4,5)}f(1,0,1,1) = 1$
$u_{54} = {}^{(4,5)}f(1,0,0,1) = 0$	$u_{163} = {}^{(4,5)}f(0,1,1,1) = 0$
$u_{55} = {}^{(4,5)}f(0,0,1,1) = 0$	$u_{164} = {}^{(4,5)}f(1,1,1,1) = 1$ **−(41)**
$u_{56} = {}^{(4,5)}f(0,1,1,1) = 0$ **−(14)**	$u_{165} = {}^{(4,5)}f(1,1,1,1) = 1$
$u_{57} = {}^{(4,5)}f(1,1,1,0) = 0$ **−(40)**	$u_{166} = {}^{(4,5)}f(1,1,1,1) = 1$
$u_{58} = {}^{(4,5)}f(1,1,0,1) = 0$ **−(38)**	$u_{167} = {}^{(4,5)}f(1,1,1,1) = 1$
$u_{59} = {}^{(4,5)}f(1,0,1,1) = 1$ **−(32)**	$u_{168} = {}^{(4,5)}f(1,1,1,2) = 2$ **−(42)**
$u_{60} = {}^{(4,5)}f(0,1,1,2) = 1$ **−(15)**	$u_{169} = {}^{(4,5)}f(1,1,2,1) = 1$ **−(44)**
$u_{61} = {}^{(4,5)}f(1,1,2,0) = 0$ **−(43)**	$u_{170} = {}^{(4,5)}f(1,2,1,1) = 0$ **−(50)**
$u_{62} = {}^{(4,5)}f(1,2,0,1) = 2$ **−(47)**	$u_{171} = {}^{(4,5)}f(2,1,1,2) = 0$ **−(69)**
$u_{63} = {}^{(4,5)}f(2,0,1,2) = 0$ **−(60)**	$u_{172} = {}^{(4,5)}f(1,1,2,0) = 0$
$u_{64} = {}^{(4,5)}f(0,1,2,0) = 2$	$u_{173} = {}^{(4,5)}f(1,2,0,1) = 2$
$u_{65} = {}^{(4,5)}f(1,2,0,0) = 1$	$u_{174} = {}^{(4,5)}f(2,0,1,1) = 2$
$u_{66} = {}^{(4,5)}f(2,0,0,1) = 1$	$u_{175} = {}^{(4,5)}f(0,1,1,2) = 1$
$u_{67} = {}^{(4,5)}f(0,0,1,2) = 1$	$u_{176} = {}^{(4,5)}f(1,1,2,1) = 1$
$u_{68} = {}^{(4,5)}f(0,1,2,1) = 0$ **−(17)**	$u_{177} = {}^{(4,5)}f(1,2,1,1) = 0$
$u_{69} = {}^{(4,5)}f(1,2,1,0) = 2$ **−(49)**	$u_{178} = {}^{(4,5)}f(2,1,1,1) = 2$
$u_{70} = {}^{(4,5)}f(2,1,0,1) = 1$ **−(65)**	$u_{179} = {}^{(4,5)}f(1,1,1,2) = 2$
$u_{71} = {}^{(4,5)}f(1,0,1,2) = 2$ **−(33)**	$u_{180} = {}^{(4,5)}f(1,1,2,2) = 2$ **−(45)**
$u_{72} = {}^{(4,5)}f(0,1,2,2) = 1$ **−(18)**	$u_{181} = {}^{(4,5)}f(1,2,2,1) = 0$ **−(53)**
$u_{73} = {}^{(4,5)}f(1,2,2,0) = 2$ **−(52)**	$u_{182} = {}^{(4,5)}f(2,2,1,2) = 2$ **−(78)**
$u_{74} = {}^{(4,5)}f(2,2,0,2) = 1$ **−(75)**	$u_{183} = {}^{(4,5)}f(2,1,2,0) = 1$

$u_{75} = {}^{(4,5)}f(2,0,2,0) = 1$ **−(61)**	$u_{184} = {}^{(4,5)}f(1,2,0,0) = 1$
$u_{76} = {}^{(4,5)}f(0,2,0,0) = 0$	$u_{185} = {}^{(4,5)}f(2,0,0,1) = 1$
$u_{77} = {}^{(4,5)}f(2,0,0,0) = 0$	$u_{186} = {}^{(4,5)}f(0,0,1,2) = 1$
$u_{78} = {}^{(4,5)}f(0,0,0,2) = 0$	$u_{187} = {}^{(4,5)}f(0,1,2,0) = 2$
$u_{79} = {}^{(4,5)}f(0,0,2,0) = 2$	$u_{188} = {}^{(4,5)}f(1,2,0,1) = 2$
$u_{80} = {}^{(4,5)}f(0,2,0,1) = 1$	$u_{189} = {}^{(4,5)}f(2,0,1,1) = 2$
$u_{81} = {}^{(4,5)}f(2,0,1,0) = 1$	$u_{190} = {}^{(4,5)}f(0,1,1,2) = 1$
$u_{82} = {}^{(4,5)}f(0,1,0,2) = 0$	$u_{191} = {}^{(4,5)}f(1,1,2,0) = 0$
$u_{83} = {}^{(4,5)}f(1,0,2,0) = 0$	$u_{192} = {}^{(4,5)}f(1,2,0,2) = 0$
$u_{84} = {}^{(4,5)}f(0,2,0,2) = 2$ **−(21)**	$u_{193} = {}^{(4,5)}f(2,0,2,1) = 2$
$u_{85} = {}^{(4,5)}f(2,0,2,0) = 1$	$u_{194} = {}^{(4,5)}f(0,2,1,2) = 0$
$u_{86} = {}^{(4,5)}f(0,2,0,2) = 2$	$u_{195} = {}^{(4,5)}f(2,1,2,1) = 2$ **−(71)**
$u_{87} = {}^{(4,5)}f(2,0,2,1) = 2$ **−(62)**	$u_{196} = {}^{(4,5)}f(1,2,1,0) = 2$
$u_{88} = {}^{(4,5)}f(0,2,1,0) = 1$	$u_{197} = {}^{(4,5)}f(2,1,0,1) = 1$
$u_{89} = {}^{(4,5)}f(2,1,0,0) = 0$	$u_{198} = {}^{(4,5)}f(1,0,1,2) = 2$
$u_{90} = {}^{(4,5)}f(1,0,0,2) = 1$	$u_{199} = {}^{(4,5)}f(0,1,2,1) = 0$
$u_{91} = {}^{(4,5)}f(0,0,2,1) = 0$	$u_{200} = {}^{(4,5)}f(1,2,1,1) = 0$
$u_{92} = {}^{(4,5)}f(0,2,1,1) = 2$ **−(23)**	$u_{201} = {}^{(4,5)}f(2,1,1,1) = 2$
$u_{93} = {}^{(4,5)}f(2,1,1,0) = 1$ **−(67)**	$u_{202} = {}^{(4,5)}f(1,1,1,2) = 2$
$u_{94} = {}^{(4,5)}f(1,1,0,2) = 1$ **−(39)**	$u_{203} = {}^{(4,5)}f(1,1,2,1) = 1$
$u_{95} = {}^{(4,5)}f(1,0,2,1) = 1$ **−(35)**	$u_{204} = {}^{(4,5)}f(1,2,1,2) = 1$ **−(51)**
$u_{96} = {}^{(4,5)}f(0,2,1,2) = 0$ **−(24)**	$u_{205} = {}^{(4,5)}f(2,1,2,1) = 2$
$u_{97} = {}^{(4,5)}f(2,1,2,0) = 1$ **−(70)**	$u_{206} = {}^{(4,5)}f(1,2,1,2) = 1$
$u_{98} = {}^{(4,5)}f(1,2,0,2) = 0$ **−(48)**	$u_{207} = {}^{(4,5)}f(2,1,2,2) = 0$ **−(72)**
$u_{99} = {}^{(4,5)}f(2,0,2,2) = 0$ **−(63)**	$u_{208} = {}^{(4,5)}f(1,2,2,0) = 2$
$u_{100} = {}^{(4,5)}f(0,2,2,0) = 1$	$u_{209} = {}^{(4,5)}f(2,2,0,1) = 0$
$u_{101} = {}^{(4,5)}f(2,2,0,0) = 2$ **−(73)**	$u_{210} = {}^{(4,5)}f(2,0,1,2) = 0$
$u_{102} = {}^{(4,5)}f(2,0,0,2) = 2$	$u_{211} = {}^{(4,5)}f(0,1,2,2) = 1$
$u_{103} = {}^{(4,5)}f(0,0,2,2) = 1$	$u_{212} = {}^{(4,5)}f(1,2,2,1) = 0$
$u_{104} = {}^{(4,5)}f(0,2,2,1) = 2$ **−(26)**	$u_{213} = {}^{(4,5)}f(2,2,1,1) = 1$
$u_{105} = {}^{(4,5)}f(2,2,1,0) = 0$ **−(76)**	$u_{214} = {}^{(4,5)}f(2,1,1,2) = 0$
$u_{106} = {}^{(4,5)}f(2,1,0,2) = 2$ **−(66)**	$u_{215} = {}^{(4,5)}f(1,1,2,2) = 2$
$u_{107} = {}^{(4,5)}f(1,0,2,2) = 2$ **−(36)**	$u_{216} = {}^{(4,5)}f(1,2,2,2) = 1$ **−(54)**
$u_{108} = {}^{(4,5)}f(0,2,2,2) = 0$ **−(27)**	$u_{217} = {}^{(4,5)}f(2,2,2,2) = 2$ **−(81)**
$u_{109} = {}^{(4,5)}f(2,2,2,1) = 1$ **−(80)**	$u_{218} = {}^{(4,5)}f(2,2,2,0) = 0$ **−(79)**

Таблица А3.16. Функция дешифрования $^{(4,5)}f$ для Примера 4.1.2

№	Значение	№	Значение	№	Значение
(1)	$^{(4,5)}f(0,0,0,0) = 1$	(19)	$^{(4,5)}f(1,0,0,0) = 2$	(55)	$^{(4,5)}f(2,0,0,0) = 0$
(2)	$^{(4,5)}f(0,0,0,1) = 2$	(20)	$^{(4,5)}f(1,0,0,1) = 0$	(56)	$^{(4,5)}f(2,0,0,1) = 1$
(3)	$^{(4,5)}f(0,0,0,2) = 0$	(21)	$^{(4,5)}f(1,0,0,2) = 1$	(57)	$^{(4,5)}f(2,0,0,2) = 2$
(4)	$^{(4,5)}f(0,0,1,0) = 2$	(22)	$^{(4,5)}f(1,0,1,0) = 0$	(58)	$^{(4,5)}f(2,0,1,0) = 1$
(5)	$^{(4,5)}f(0,0,1,1) = 0$	(23)	$^{(4,5)}f(1,0,1,1) = 1$	(59)	$^{(4,5)}f(2,0,1,1) = 2$
(6)	$^{(4,5)}f(0,0,1,2) = 1$	(24)	$^{(4,5)}f(1,0,1,2) = 2$	(60)	$^{(4,5)}f(2,0,1,2) = 0$
(7)	$^{(4,5)}f(0,0,2,0) = 2$	(25)	$^{(4,5)}f(1,0,2,0) = 0$	(61)	$^{(4,5)}f(2,0,2,0) = 1$
(8)	$^{(4,5)}f(0,0,2,1) = 0$	(26)	$^{(4,5)}f(1,0,2,1) = 1$	(62)	$^{(4,5)}f(2,0,2,1) = 2$
(9)	$^{(4,5)}f(0,0,2,2) = 1$	(27)	$^{(4,5)}f(1,0,2,2) = 2$	(63)	$^{(4,5)}f(2,0,2,2) = 0$
(10)	$^{(4,5)}f(0,1,0,0) = 1$	(28)	$^{(4,5)}f(1,1,0,0) = 2$	(64)	$^{(4,5)}f(2,1,0,0) = 0$
(11)	$^{(4,5)}f(0,1,0,1) = 2$	(29)	$^{(4,5)}f(1,1,0,1) = 0$	(65)	$^{(4,5)}f(2,1,0,1) = 1$
(12)	$^{(4,5)}f(0,1,0,2) = 0$	(30)	$^{(4,5)}f(1,1,0,2) = 1$	(66)	$^{(4,5)}f(2,1,0,2) = 2$
(13)	$^{(4,5)}f(0,1,1,0) = 2$	(31)	$^{(4,5)}f(1,1,1,0) = 0$	(67)	$^{(4,5)}f(2,1,1,0) = 1$
(14)	$^{(4,5)}f(0,1,1,1) = 0$	(32)	$^{(4,5)}f(1,1,1,1) = 1$	(68)	$^{(4,5)}f(2,1,1,1) = 2$
(15)	$^{(4,5)}f(0,1,1,2) = 1$	(33)	$^{(4,5)}f(1,1,1,2) = 2$	(69)	$^{(4,5)}f(2,1,1,2) = 0$
(16)	$^{(4,5)}f(0,1,2,0) = 2$	(34)	$^{(4,5)}f(1,1,2,0) = 0$	(70)	$^{(4,5)}f(2,1,2,0) = 1$
(17)	$^{(4,5)}f(0,1,2,1) = 0$	(35)	$^{(4,5)}f(1,1,2,1) = 1$	(71)	$^{(4,5)}f(2,1,2,1) = 2$
(18)	$^{(4,5)}f(0,1,2,2) = 1$	(36)	$^{(4,5)}f(1,1,2,2) = 2$	(72)	$^{(4,5)}f(2,1,2,2) = 0$
(19)	$^{(4,5)}f(0,2,0,0) = 0$	(37)	$^{(4,5)}f(1,2,0,0) = 1$	(73)	$^{(4,5)}f(2,2,0,0) = 2$
(20)	$^{(4,5)}f(0,2,0,1) = 1$	(38)	$^{(4,5)}f(1,2,0,1) = 2$	(74)	$^{(4,5)}f(2,2,0,1) = 0$
(21)	$^{(4,5)}f(0,2,0,2) = 2$	(39)	$^{(4,5)}f(1,2,0,2) = 0$	(75)	$^{(4,5)}f(2,2,0,2) = 1$
(22)	$^{(4,5)}f(0,2,1,0) = 1$	(40)	$^{(4,5)}f(1,2,1,0) = 2$	(76)	$^{(4,5)}f(2,2,1,0) = 0$
(23)	$^{(4,5)}f(0,2,1,1) = 2$	(41)	$^{(4,5)}f(1,2,1,1) = 0$	(77)	$^{(4,5)}f(2,2,1,1) = 1$
(24)	$^{(4,5)}f(0,2,1,2) = 0$	(42)	$^{(4,5)}f(1,2,1,2) = 1$	(78)	$^{(4,5)}f(2,2,1,2) = 2$
(25)	$^{(4,5)}f(0,2,2,0) = 1$	(43)	$^{(4,5)}f(1,2,2,0) = 2$	(79)	$^{(4,5)}f(2,2,2,0) = 0$
(26)	$^{(4,5)}f(0,2,2,1) = 2$	(44)	$^{(4,5)}f(1,2,2,1) = 0$	(80)	$^{(4,5)}f(2,2,2,1) = 1$
(27)	$^{(4,5)}f(0,2,2,2) = 0$	(45)	$^{(4,5)}f(1,2,2,2) = 1$	(81)	$^{(4,5)}f(2,2,2,2) = 2$

Таблица А3.17. Функция шифрования f для Примера 4.1.2

№	Значение	№	Значение	№	Значение
(1)	$f(0,0,0,0) = 2$	(28)	$f(1,0,0,0) = 1$	(55)	$f(2,0,0,0) = 0$
(2)	$f(0,0,0,1) = 0$	(29)	$f(1,0,0,1) = 2$	(56)	$f(2,0,0,1) = 1$
(3)	$f(0,0,0,2) = 1$	(30)	$f(1,0,0,2) = 0$	(57)	$f(2,0,0,2) = 2$
(4)	$f(0,0,1,0) = 1$	(31)	$f(1,0,1,0) = 0$	(58)	$f(2,0,1,0) = 2$
(5)	$f(0,0,1,1) = 2$	(32)	$f(1,0,1,1) = 1$	(59)	$f(2,0,1,1) = 0$
(6)	$f(0,0,1,2) = 0$	(33)	$f(1,0,1,2) = 2$	(60)	$f(2,0,1,2) = 1$
(7)	$f(0,0,2,0) = 1$	(34)	$f(1,0,2,0) = 0$	(61)	$f(2,0,2,0) = 2$
(8)	$f(0,0,2,1) = 2$	(35)	$f(1,0,2,1) = 1$	(62)	$f(2,0,2,1) = 0$
(9)	$f(0,0,2,2) = 0$	(36)	$f(1,0,2,2) = 2$	(63)	$f(2,0,2,2) = 1$
(10)	$f(0,1,0,0) = 2$	(37)	$f(1,1,0,0) = 1$	(64)	$f(2,1,0,0) = 0$
(11)	$f(0,1,0,1) = 0$	(38)	$f(1,1,0,1) = 2$	(65)	$f(2,1,0,1) = 1$
(12)	$f(0,1,0,2) = 1$	(39)	$f(1,1,0,2) = 0$	(66)	$f(2,1,0,2) = 2$
(13)	$f(0,1,1,0) = 1$	(40)	$f(1,1,1,0) = 0$	(67)	$f(2,1,1,0) = 2$
(14)	$f(0,1,1,1) = 2$	(41)	$f(1,1,1,1) = 1$	(68)	$f(2,1,1,1) = 0$
(15)	$f(0,1,1,2) = 0$	(42)	$f(1,1,1,2) = 2$	(69)	$f(2,1,1,2) = 1$
(16)	$f(0,1,2,0) = 1$	(43)	$f(1,1,2,0) = 0$	(70)	$f(2,1,2,0) = 2$
(17)	$f(0,1,2,1) = 2$	(44)	$f(1,1,2,1) = 1$	(71)	$f(2,1,2,1) = 0$
(18)	$f(0,1,2,2) = 0$	(45)	$f(1,1,2,2) = 2$	(72)	$f(2,1,2,2) = 1$
(19)	$f(0,2,0,0) = 0$	(46)	$f(1,2,0,0) = 2$	(73)	$f(2,2,0,0) = 1$
(20)	$f(0,2,0,1) = 1$	(47)	$f(1,2,0,1) = 0$	(74)	$f(2,2,0,1) = 2$
(21)	$f(0,2,0,2) = 2$	(48)	$f(1,2,0,2) = 1$	(75)	$f(2,2,0,2) = 0$
(22)	$f(0,2,1,0) = 2$	(49)	$f(1,2,1,0) = 1$	(76)	$f(2,2,1,0) = 0$
(23)	$f(0,2,1,1) = 0$	(50)	$f(1,2,1,1) = 2$	(77)	$f(2,2,1,1) = 1$
(24)	$f(0,2,1,2) = 1$	(51)	$f(1,2,1,2) = 0$	(78)	$f(2,2,1,2) = 2$
(25)	$f(0,2,2,0) = 2$	(52)	$f(1,2,2,0) = 1$	(79)	$f(2,2,2,0) = 0$
(26)	$f(0,2,2,1) = 0$	(53)	$f(1,2,2,1) = 2$	(80)	$f(2,2,2,1) = 1$
(27)	$f(0,2,2,2) = 1$	(54)	$f(1,2,2,2) = 0$	(81)	$f(2,2,2,2) = 2$

Таблица А3.18. Процесс шифрования для Примера 4.2.2

$v_1 = f(l_1, l_2, l_3, u_1) =$ $= f(1,0,0,0) = 1$	$v_{114} = f(0,1,1,0) = 1$	$v_{227} = f(0,0,0,0) = 2$
$v_2 = f(l_4, l_5, l_6, u_2) =$ $= f(2,1,1,0) = 2$	$v_{115} = f(1,1,1,0) = 0$	$v_{228} = f(0,0,2,2) = 0$
$v_3 = f(l_7, l_8, l_9, u_3) =$ $= f(0,0,0,0) = 2$	$v_{116} = f(1,1,0,1) = 2 - (\mathbf{38})$	$v_{229} = f(0,2,0,2) = 2$
$v_4 = f(v_1, v_2, v_3, u_4) =$ $= f(1,2,2,0) = 1 -$**(52)**	$v_{117} = f(1,0,2,1) = 1 -$**(35)**	$v_{230} = f(2,0,2,0) = 2$
$v_5 = f(2,2,1,0) = 0 -$**(76)**	$v_{118} = f(0,2,1,0) = 2 -$**(22)**	$v_{231} = f(0,2,2,1) = 0$
$v_6 = f(2,1,0,0) = 0 -$**(64)**	$v_{119} = f(2,1,2,0) = 2 -$**(70)**	$v_{232} = f(2,2,0,0) = 1$
$v_7 = f(1,0,0,0) = 1 -$**(28)**	$v_{120} = f(1,2,2,2) = 0$	$v_{233} = f(2,0,1,2) = 1$
$v_8 = f(0,0,1,1) = 2 -$**(5)**	$v_{121} = f(2,2,0,1) = 2 -$**(74)**	$v_{234} = f(0,1,1,0) = 1$
$v_9 = f(0,1,2,0) = 1 -$**(16)**	$v_{122} = f(2,0,2,0) = 2$	$v_{235} = f(1,1,1,1) = 1$
$v_{10} = f(1,2,1,0) = 1 -$**(49)**	$v_{123} = f(0,2,2,1) = 0$	$v_{236} = f(1,1,1,1) = 1$
$v_{11} = f(2,1,1,0) = 2 -$**(67)**	$v_{124} = f(2,2,0,0) = 1$	$v_{237} = f(1,1,1,2) = 2$
$v_{12} = f(1,1,2,2) = 2 -$**(45)**	$v_{125} = f(2,0,1,1) = 0$	$v_{238} = f(1,1,2,0) = 0$
$v_{13} = f(1,2,2,0) = 1$	$v_{126} = f(0,1,0,0) = 2$	$v_{239} = f(1,2,0,1) = 0$
$v_{14} = f(2,2,1,0) = 0$	$v_{127} = f(1,0,2,1) = 1$	$v_{240} = f(2,0,0,2) = 2$
$v_{15} = f(2,1,0,1) = 1 -$**(65)**	$v_{128} = f(0,2,1,1) = 0 -$**(23)**	$v_{241} = f(0,0,2,2) = 0$
$v_{16} = f(1,0,1,0) = 0 -$**(31)**	$v_{129} = f(2,1,0,1) = 1$	$v_{242} = f(0,2,0,0) = 0$
$v_{17} = f(0,1,0,0) = 2 -$**(10)**	$v_{130} = f(1,0,1,0) = 0$	$v_{243} = f(2,0,0,2) = 2$
$v_{18} = f(1,0,2,0) = 0 -$**(34)**	$v_{131} = f(0,1,0,1) = 0 -$**(11)**	$v_{244} = f(0,0,2,0) = 1$
$v_{19} = f(0,2,0,1) = 1 -$**(20)**	$v_{132} = f(1,0,0,2) = 0 -$**(30)**	$v_{245} = f(0,2,1,2) = 1$
$v_{20} = f(2,0,1,1) = 0 -$**(59)**	$v_{133} = f(0,0,0,1) = 0 -$**(2)**	$v_{246} = f(2,1,1,0) = 2$
$v_{21} = f(0,1,0,0) = 2$	$v_{134} = f(0,0,0,0) = 2 -$**(1)**	$v_{247} = f(1,1,2,2) = 2$
$v_{22} = f(1,0,2,0) = 0$	$v_{135} = f(0,0,2,2) = 0 -$**(9)**	$v_{248} = f(1,2,2,1) = 2$
$v_{23} = f(0,2,0,1) = 1$	$v_{136} = f(0,2,0,0) = 0$	$v_{249} = f(2,2,2,2) = 2$
$v_{24} = f(2,0,1,2) = 1 -$**(60)**	$v_{137} = f(2,0,0,1) = 1$	$v_{250} = f(2,2,2,0) = 0$
$v_{25} = f(0,1,1,0) = 1 -$**(13)**	$v_{138} = f(0,0,1,0) = 1$	$v_{251} = f(2,2,0,2) = 0$
$v_{26} = f(1,1,1,0) = 0 -$**(40)**	$v_{139} = f(0,1,1,2) = 0 - (\mathbf{15})$	$v_{252} = f(2,0,0,2) = 2$
$v_{27} = f(1,1,0,2) = 0 -$**(39)**	$v_{140} = f(1,1,0,1) = 2$	$v_{253} = f(0,0,2,2) = 0$
$v_{28} = f(1,0,0,0) = 1$	$v_{141} = f(1,0,2,1) = 1$	$v_{254} = f(0,2,0,1) = 1$
$v_{29} = f(0,0,1,0) = 1 -$**(4)**	$v_{142} = f(0,2,1,0) = 2$	$v_{255} = f(2,0,1,0) = 2$
$v_{30} = f(0,1,1,0) = 1$	$v_{143} = f(2,1,2,2) = 1 - (\mathbf{72})$	$v_{256} = f(0,1,2,0) = 1$
$v_{31} = f(1,1,1,2) = 2 -$**(42)**	$v_{144} = f(1,2,1,2) = 0 -$**(51)**	$v_{257} = f(1,2,1,2) = 0$
$v_{32} = f(1,1,2,1) = 1 -$**(44)**	$v_{145} = f(2,1,0,1) = 1$	$v_{258} = f(2,1,0,1) = 1$
$v_{33} = f(1,2,1,0) = 1$	$v_{146} = f(1,0,1,1) = 1$	$v_{259} = f(1,0,1,0) = 0$
$v_{34} = f(2,1,1,0) = 2$	$v_{147} = f(0,1,1,0) = 1$	$v_{260} = f(0,1,0,1) = 0$
$v_{35} = f(1,1,2,2) = 2$	$v_{148} = f(1,1,1,0) = 0$	$v_{261} = f(1,0,0,2) = 0$
$v_{36} = f(1,2,2,2) = 0 -$**(54)**	$v_{149} = f(1,1,0,1) = 2$	$v_{262} = f(0,0,0,1) = 0$

$v_{37} = f(2,2,0,0) = 1 -$**(73)**	$v_{150} = f(1,0,2,1) = 1$	$v_{263} = f(0,0,0,0) = 2$
$v_{38} = f(2,0,1,1) = 0$	$v_{151} = f(0,2,1,0) = 2$	$v_{264} = f(0,0,2,2) = 0$
$v_{39} = f(0,1,0,0) = 2$	$v_{152} = f(2,1,2,1) = 0$	$v_{265} = f(0,2,0,2) = 2$
$v_{40} = f(1,0,2,0) = 0$	$v_{153} = f(1,2,0,1) = 0$	$v_{266} = f(2,0,2,1) = 0$
$v_{41} = f(0,2,0,0) = 0 -$**(19)**	$v_{154} = f(2,0,0,1) = 1$	$v_{267} = f(0,2,0,1) = 1$
$v_{42} = f(2,0,0,1) = 1 -$**(56)**	$v_{155} = f(0,0,1,0) = 1$	$v_{268} = f(2,0,1,0) = 2$
$v_{43} = f(0,0,1,0) = 1$	$v_{156} = f(0,1,1,2) = 0$	$v_{269} = f(0,1,2,2) = 0$
$v_{44} = f(0,1,1,1) = 2 - (\mathbf{14})$	$v_{157} = f(1,1,0,1) = 2$	$v_{270} = f(1,2,0,1) = 0$
$v_{45} = f(1,1,2,0) = 0 -$**(43)**	$v_{158} = f(1,0,2,1) = 1$	$v_{271} = f(2,0,0,1) = 1$
$v_{46} = f(1,2,0,1) = 0 - (\mathbf{47})$	$v_{159} = f(0,2,1,1) = 0$	$v_{272} = f(0,0,1,1) = 2$
$v_{47} = f(2,0,0,0) = 0 -$**(55)**	$v_{160} = f(2,1,0,0) = 0$	$v_{273} = f(0,1,2,2) = 0$
$v_{48} = f(0,0,0,2) = 1 - (\mathbf{3})$	$v_{161} = f(1,0,01) = 2 - (\mathbf{29})$	$v_{274} = f(1,2,0,1) = 0$
$v_{49} = f(0,0,1,0) = 1$	$v_{162} = f(0,0,2,1) = 2 -$**(8)**	$v_{275} = f(2,0,0,1) = 1$
$v_{50} = f(0,1,1,1) = 2$	$v_{163} = f(0,2,2,1) = 0$	$v_{276} = f(0,0,1,2) = 0$
$v_{51} = f(1,1,2,1) = 1$	$v_{164} = f(2,2,0,1) = 2$	$v_{277} = f(0,1,0,2) = 1$
$v_{52} = f(1,2,1,0) = 1$	$v_{165} = f(2,0,2,1) = 0$	$v_{278} = f(1,0,1,1) = 1$
$v_{53} = f(2,1,1,0) = 2$	$v_{166} = f(0,2,0,1) = 1$	$v_{279} = f(0,1,1,2) = 0$
$v_{54} = f(1,1,2,1) = 1$	$v_{167} = f(2,0,1,1) = 0$	$v_{280} = f(1,1,0,0) = 1$
$v_{55} = f(1,2,1,1) = 2 - (\mathbf{50})$	$v_{168} = f(0,1,0,2) = 1$	$v_{281} = f(1,0,1,2) = 2$
$v_{56} = f(2,1,2,1) = 0 - (\mathbf{71})$	$v_{169} = f(1,0,1,1) = 1$	$v_{282} = f(0,1,2,1) = 2$
$v_{57} = f(1,2,0,0) = 2 - (\mathbf{46})$	$v_{170} = f(0,1,1,1) = 2$	$v_{283} = f(1,2,2,2) = 0$
$v_{58} = f(2,0,2,1) = 0 - (\mathbf{62})$	$v_{171} = f(1,1,2,2) = 2$	$v_{284} = f(2,2,0,1) = 2$
$v_{59} = f(0,2,0,1) = 1$	$v_{172} = f(1,2,2,0) = 1$	$v_{285} = f(2,0,2,2) = 1$
$v_{60} = f(2,0,1,2) = 1$	$v_{173} = f(2,2,1,1) = 1 -$**(77)**	$v_{286} = f(0,2,1,1) = 0$
$v_{61} = f(0,1,1,0) = 1$	$v_{174} = f(2,1,1,1) = 0 -$**(68)**	$v_{287} = f(2,1,0,2) = 2$
$v_{62} = f(1,1,1,1) = 1 -$**(41)**	$v_{175} = f(1,1,0,2) = 0$	$v_{288} = f(1,0,2,2) = 2 -$**(36)**
$v_{63} = f(1,1,1,2) = 2$	$v_{176} = f(1,0,0,1) = 2$	$v_{289} = f(0,2,2,2) = 1$
$v_{64} = f(1,1,2,0) = 0$	$v_{177} = f(0,0,2,1) = 2$	$v_{290} = f(2,2,1,2) = 2 -$**(78)**
$v_{65} = f(1,2,0,0) = 2$	$v_{178} = f(0,2,2,1) = 0$	$v_{291} = f(2,1,2,0) = 2$
$v_{66} = f(2,0,2,1) = 0$	$v_{179} = f(2,2,0,2) = 0 -$**(75)**	$v_{292} = f(1,2,2,0) = 1$
$v_{67} = f(0,2,0,2) = 2 - (\mathbf{21})$	$v_{180} = f(2,0,0,2) = 2$	$v_{293} = f(2,2,1,2) = 2$
$v_{68} = f(2,0,2,1) = 0$	$v_{181} = f(0,0,2,1) = 2$	$v_{294} = f(2,1,2,2) = 1$
$v_{69} = f(0,2,0,0) = 0$	$v_{182} = f(0,2,2,2) = 1 -$**(27)**	$v_{295} = f(1,2,1,0) = 1$
$v_{70} = f(2,0,0,1) = 1$	$v_{183} = f(2,2,1,0) = 0$	$v_{296} = f(2,1,1,1) = 0$
$v_{71} = f(0,0,1,2) = 0 - (\mathbf{6})$	$v_{184} = f(2,1,0,0) = 0$	$v_{297} = f(1,1,0,2) = 0$
$v_{72} = f(0,1,0,2) = 1 -$**(12)**	$v_{185} = f(1,0,0,1) = 2$	$v_{298} = f(1,0,0,2) = 0$
$v_{73} = f(1,0,1,0) = 0$	$v_{186} = f(0,0,2,2) = 0$	$v_{299} = f(0,0,0,0) = 2$
$v_{74} = f(0,1,0,2) = 1$	$v_{187} = f(0,2,0,0) = 0$	$v_{300} = f(0,0,2,2) = 0$
$v_{75} = f(1,0,1,0) = 0$	$v_{188} = f(2,0,0,1) = 1$	$v_{301} = f(0,2,0,2) = 2$
$v_{76} = f(0,1,0,0) = 2$	$v_{189} = f(0,0,1,1) = 2$	$v_{302} = f(2,0,2,2) = 1$
$v_{77} = f(1,0,2,0) = 0$	$v_{190} = f(0,1,2,2) = 0$	$v_{303} = f(0,2,1,1) = 0$

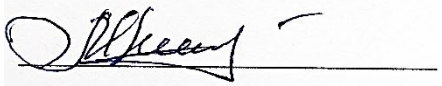
$v_{78} = f(0,2,0,2) = 2$	$v_{191} = f(1,2,0,0) = 2$	$v_{304} = f(2,1,0,0) = 0$
$v_{79} = f(2,0,2,0) = 2 -$**(61)**	$v_{192} = f(2,0,2,2) = 1 - (\mathbf{63})$	$v_{305} = f(1,0,0,2) = 0$
$v_{80} = f(0,2,2,1) = 0 - (\mathbf{26})$	$v_{193} = f(0,2,1,1) = 0$	$v_{306} = f(0,0,0,2) = 1$
$v_{81} = f(2,2,0,0) = 1$	$v_{194} = f(2,1,0,2) = 2 -$**(66)**	$v_{307} = f(0,0,1,1) = 2$
$v_{82} = f(2,0,1,2) = 1$	$v_{195} = f(1,0,2,1) = 1$	$v_{308} = f(0,1,2,1) = 2$
$v_{83} = f(0,1,1,0) = 1$	$v_{196} = f(0,2,1,0) = 2$	$v_{309} = f(1,2,2,2) = 0$
$v_{84} = f(1,1,1,2) = 2$	$v_{197} = f(2,1,2,1) = 0$	$v_{310} = f(2,2,0,2) = 0$
$v_{85} = f(1,1,2,0) = 0$	$v_{198} = f(1,2,0,2) = 1$	$v_{311} = f(2,0,0,1) = 1$
$v_{86} = f(1,2,0,2) = 1 -$**(48)**	$v_{199} = f(2,0,1,1) = 0$	$v_{312} = f(0,0,1,2) = 0$
$v_{87} = f(2,0,1,1) = 0$	$v_{200} = f(0,1,0,1) = 0$	$v_{313} = f(0,1,0,2) = 1$
$v_{88} = f(0,1,0,0) = 2$	$v_{201} = f(1,0,0,1) = 2$	$v_{314} = f(1,0,1,2) = 2$
$v_{89} = f(1,0,2,0) = 0$	$v_{202} = f(0,0,2,2) = 0$	$v_{315} = f(0,1,2,2) = 0$
$v_{90} = f(0,2,0,2) = 2$	$v_{203} = f(0,2,0,1) = 1$	$v_{316} = f(1,2,0,0) = 2$
$v_{91} = f(2,0,2,1) = 0$	$v_{204} = f(2,0,1,2) = 1$	$v_{317} = f(2,0,2,2) = 1$
$v_{92} = f(0,2,0,1) = 1$	$v_{205} = f(0,1,1,1) = 2$	$v_{318} = f(0,2,1,2) = 1$
$v_{93} = f(2,0,1,0) = 2 - (\mathbf{58})$	$v_{206} = f(1,1,2,2) = 2$	$v_{319} = f(2,1,1,2) = 1$
$v_{94} = f(0,1,2,2) = 0 - (\mathbf{18})$	$v_{207} = f(1,2,2,2) = 0$	$v_{320} = f(1,1,1,1) = 1$
$v_{95} = f(1,2,0,1) = 0$	$v_{208} = f(2,2,0,0) = 1$	$v_{321} = f(1,1,1,2) = 2$
$v_{96} = f(2,0,0,2) = 2 -$**(57)**	$v_{209} = f(2,0,1,1) = 0$	$v_{322} = f(1,1,2,2) = 2$
$v_{97} = f(0,0,2,0) = 1 - (\mathbf{7})$	$v_{210} = f(0,1,0,2) = 1$	$v_{323} = f(1,2,2,2) = 0$
$v_{98} = f(0,2,1,2) = 1 - (\mathbf{24})$	$v_{211} = f(1,0,1,2) = 2$	$v_{324} = f(2,2,0,2) = 0$
$v_{99} = f(2,1,1,2) = 1 - (\mathbf{69})$	$v_{212} = f(0,1,2,1) = 2 -$**(17)**	$v_{325} = f(2,0,0,0) = 0$
$v_{100} = f(1,1,1,0) = 0$	$v_{213} = f(1,2,2,1) = 2 -$**(53)**	$v_{326} = f(0,0,0,0) = 2$
$v_{101} = f(1,1,0,0) = 1 -$**(37)**	$v_{214} = f(2,2,2,2) = 2 -$**(81)**	$v_{327} = f(0,0,2,0) = 1$
$v_{102} = f(1,0,1,2) = 2 -$**(33)**	$v_{215} = f(2,2,2,2) = 2$	$v_{328} = f(0,2,1,0) = 2$
$v_{103} = f(0,1,2,2) = 0$	$v_{216} = f(2,2,2,2) = 2$	$v_{329} = f(2,1,2,0) = 2$
$v_{104} = f(1,2,0,1) = 0$	$v_{217} = f(2,2,2,2) = 2$	$v_{330} = f(1,2,2,0) = 1$
$v_{105} = f(2,0,0,0) = 0$	$v_{218} = f(2,2,2,0) = 0 -$**(79)**	$v_{331} = f(2,2,1,0) = 0$
$v_{106} = f(0,0,0,2) = 1$	$v_{219} = f(2,2,0,0) = 1$	$v_{332} = f(2,1,0,1) = 1$
$v_{107} = f(0,0,1,2) = 0$	$v_{220} = f(2,0,1,0) = 2$	$v_{333} = f(1,0,1,0) = 0$
$v_{108} = f(0,1,0,2) = 1$	$v_{221} = f(0,1,2,2) = 0$	$v_{334} = f(0,1,0,0) = 2$
$v_{109} = f(1,0,1,1) = 1 - (\mathbf{32})$	$v_{222} = f(1,2,0,0) = 2$	$v_{335} = f(1,0,2,0) = 0$
$v_{110} = f(0,1,1,0) = 1$	$v_{223} = f(2,0,2,0) = 2$	$v_{336} = f(0,2,0,2) = 2$
$v_{111} = f(1,1,1,0) = 0$	$v_{224} = f(0,2,2,1) = 0$	$v_{337} = f(2,0,2,0) = 2$
$v_{112} = f(1,1,0,0) = 1$	$v_{225} = f(2,2,0,2) = 0$	$v_{338} = f(0,2,2,0) = 2 -$**(25)**
$v_{113} = f(1,0,1,1) = 1$	$v_{226} = f(2,0,0,0) = 0$	

ДЕКЛАРАЦИЯ ОБ ОТВЕТСТВЕННОСТИ

Нижеподписавшийся, заявляю под личную ответственность, что материалы, представленные в докторской диссертации, являются результатом личных научных исследований и разработок. Осознаю, что в противном случае, буду нести ответственность в соответствии с действующим законодательством.

Малютина Надежда

Подпись



Дата

13.03.2023

CURRICULUM VITAE

ПЕРСОНАЛЬНЫЕ ДАННЫЕ

Малютина Надежда

Молдова, Тирасполь, ул.Карла-Либкнехта 193/91, MD-3300

0037353353781 0037377705981

23.10.03.Bab.Nadezhda@mail.ru

Пол женский | Дата рождения 21/10/1979 | Украинка по национальности

2002 – настоящее время	Старший преподаватель Приднестровского Государственного Университета имени Т.Г. Шевченко, физико-математического факультета, кафедры Алгебры, Геометрии и Методики Преподавания Математики, ул. 25 Октября, 128, Тирасполь.

ОБРАЗОВАНИЕ И ОБУЧЕНИЕ

2017-2021	Докторантура по специальности 122.03 -Модели, методы математики, программные продукты в Институте математики и информатики им. Владимира Андрунакиевича Академии наук Молдовы (в Государственном Университете Молдовы с 2021 года), г. Кишинев, Республика Молдова.
2013-2015	Магистр по специальности 080100 - Экономика, Московский институт предпринимательства и права, профиль: Экономика предприятия, Москва, Россия.
1996-2002	Высшее специалитет. Приднестровский государственный университет имени Т.Г. Шевченко, физико-математический факультет, специальность прикладная математика и информатика, Тирасполь

ЛИЧНЫЕ НАВЫКИ

Родной язык Русский

Другой язык	ПОНИМАНИЕ		РАЗГОВОРНЫЙ		ПИСЬМЕННЫЙ
	Слушание	Чтение	Разговорное общение	Разговорная постановка	
АНГЛИЙСКИЙ	B1/2	B1/2	B1/2	B1/2	B1/2

Навыки работы с компьютером
- хорошее владение инструментами Microsoft Office™
- использование LaTex

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Публикации
- Malyutina, N., Scerbacova, A., Shcherbacov, V. Markovski algorithm on i-invertibile groupoids, 2018, 3 p. [Onlain]. Available: https://arxiv.org/pdf/1806.02267.pdf.
- Malyutina, N., Cryptanalysis of some stream ciphers. In: Quasigroups and Related Systems, vol. 27, no. 2, pp. 281-292, 2019. ISSN 1561-2848.

- Malyutina, N., Cryptanalysis of some stream ciphers based on n-ary groupoids. In: Quasigroups and Related Systems, vol. 28, no. 2, pp. 251-268, 2020. ISSN 1561-2848.
- Horosh, G., Malyutina, N., Scerbacova, A., Shcherbacov, V. Units in generalized derivatives of quasigroups, 2020. [Onlain]. Available: https://arxiv.org/pdf/2009.03605.pdf.
- Horosh, G., Malyutina, N., Scherbakova, A., Şcerbacov, V. Units in generalized derivatives of quasigroups. In: Current Problems of Mathematics and Informatics, November 27-28, 2020, Chisinau, 2021, pp. 61-63. ISBN 978-9975-45-677-7.
- Malyutina, N., Shcherbacov V., An analogue of the ElGamal scheme based on the Markovski algorithm, 2021, 10 p. [online] Available: https://arxiv.org/pdf/2111.08476.pdf
- Malyutina, N., Shcherbacov V., An analogue of the ElGamal scheme based on the Markovski algorithm. In: ROMAI Journal, vol.17, no.1, 2021, pp. 105–114. ISSN (P)1841-5512; ISSN (E) 2065-7714.

Конференции

- International Conference on Mathematics, Informatics and Information Technologies dedicated to the illustrious scientist Valentin Belousov, MITI 2018, 19-21 april, Bălţi.
- Conference on Mathematical Foundations of Informatics, MFOI, Chisinau, Republic of Moldova, 2018 and 2019.
- "Tendinţe contemporane ale dezvoltării ştiinţei: viziuni ale tinerilor cercetători": Conferința Științifică a Doctoranzilor (cu participare internațională), Chișinău, 2018, 2019 (the plenary talk) and 2020.
- "International Conference "Mathematics & Information Technologies: Research and Education", Moldova, State University, Chișinău, 2019 and 2021.
- LOOPS 2019 Conference, Budapest University of Technology and Economics, July 7-July 13, 2019, Hungary.
- The 5th International Conference of Mathematical Society of the Republic of Moldova, dedicated to the 55th anniversary of the foundation of Vladimir Andrunachievici Institute of Mathematics and Computer Science (IMCS-55), September 30, 2019, Chișinău.
- Conferința științifică națională a doctoranzilor „Metodologii contemporane de cercetare și evaluare", dedicată aniversării a 75-a a USM, Chisinau, 22-23 aprilie, 2021.
- Conferinţa ştiinţifică studenţească cu participare internaţională, Chisinau, Tiraspol State University, 28 aprilie, 2021.
- Conference "Contemporary Research and Evaluation Methodologies, Dept" Chisinau, Moldova, April 22-23, 2021

Семинары

Научный семинар «Алгебра и математическая логика», посвященный памяти профессора В. Белоусова, Институт математики и информатики Академии наук Молдовы.

Звания и награды

08.17.2018 присвоено звание «Отличник народного образования».

Участия

Научный сотрудник лаборатории «Алгебра и ее приложения» Приднестровского государственного университета имени Т.Г. Шевченко (Тирасполь)