Минобрнауки России

Федеральное государственное автономное образовательное учреждение высшего образования   
«Национальный исследовательский университет   
«Московский институт электронной техники»

Факультет Микроприборов и технической кибернетики

Кафедра Высшей математики 1

Сабурова Виктория Игоревна

Бакалаврская работа   
по направлению 01.03.04 «Прикладная математика»

Критерий на открытый текст на базе решающего леса

Студент Сабурова В.И.

Научный руководитель,

к.ф.-м.н. Козлитин И.А.

Москва 2017

# Оглавление

[Оглавление 2](#_Toc480120820)

[Введение 3](#_Toc480120821)

[Глава 1 4](#_Toc480120822)

[Часть 1. Базовые понятия криптографии 4](#_Toc480120823)

[Часть 2. Изучение линейно – рекуррентных последовательностей 7](#_Toc480120824)

[Часть 3. Изучение деревьев принятия решений и рандомизированных лесов 10](#_Toc480120825)

[Глава 2 13](#_Toc480120826)

[Часть 1. Реализация процедуры потокового шифрования с помощью линейно – рекуррентных последовательностей 13](#_Toc480120827)

[1.1 Генератор потока ключей 13](#_Toc480120828)

[1.2 Принцип работы ЛРСОС. 15](#_Toc480120829)

[1.3 Критерий Вальда 18](#_Toc480120830)

[1.4 Описание работы алгоритма 20](#_Toc480120831)

[Часть 2. Построение простого критерия на открытый текст на основе принципа запрещенных биграмм 22](#_Toc480120832)

[Часть 3. Построение простого критерия на основе рандомизированного леса 23](#_Toc480120833)

[Глава 3 24](#_Toc480120834)

[Часть 1. Исследование эффективности критерия 24](#_Toc480120835)

[Заключение и выводы 25](#_Toc480120836)

[Список литературы 26](#_Toc480120837)

# Введение

В современном мире необходимость в защите информации возникла задолго до появления информационных технологий. Потребность в сокрытии передаваемой информации и ознакомления с ней лиц, которым данное сообщение не предназначалось, была реализована несколькими подходами, одному из которых и будет посвящена данная работа.   
 Подход, который заключался в преобразовании символов в набор неупорядоченных знаков, где получатель мог интерпретировать полученную информацию в осмысленное сообщение, если имел ключ к построению, позже был назван криптографией.   
 Говоря о криптографии важно упомянуть о том, что она является всего лишь небольшой частью более общей дисциплины – криптологии, науке о шифрах, которую принято разделять на две части – криптографию и криптоанализ, коим будет уделено равное внимание в работе.   
 В данной работе будут рассмотрены две гипотезы: основополагающей из них является гипотеза о том, что рассматриваемый текст, полученный в результате дешифрования, является открытым, то есть биграммы данного текста распределены в соответствии с априори известной гистограммой.

Другая гипотеза, являющаяся альтернативной, подразумевает, что получаемый на вход текст является шифрованным, то есть для него все биграммы равновероятны.   
 Для данных гипотез, соответственно, будут реализованы различные методы шифрования и дешифрования: создание ключа на основе примитивного многочлена, перебор ключей на основе последовательного анализа Вальда, критерий на основе запрещенных биграмм и т.д.

# **Глава 1**

# Часть 1. Базовые понятия криптографии

Прежде чем приступить к описанию данной работы, к конкретно поставленной задаче и ее решения введем некоторые основные понятия и определения, которые здесь используются.

Учитывая, что криптография является довольно широко используемой наукой, мы можем легко дать ей определение: криптография - наука, изучающая построение и использование систем шифрования, в том числе их стойкость, слабости и степень уязвимости относительно различных методов вскрытия, называется криптографией.

Все методы преобразования информации с целью защиты от несанкционированного доступа делятся на две группы: шифрование с закрытым ключом и шифрование с открытым ключом.

*Шифрование с открытым ключом* (ассиметричное шифрование) используется со второй половины XX века. В данную группу относятся методы шифрования, в которых для шифровки и расшифровки данных используются два разных ключа, при котором один из них (открытый ключ) может передаваться по открытому (незащищенному) каналу связи.

*Шифрование с закрытым ключом* (шифрование с секретным ключом или симметричное шифрование) используется довольно давно; в данном случае для шифрования и расшифровывания текста необходимо использовать один и тот же ключ, который обе стороны хранят в секрете от взломщиков. В данной работе представлен метод шифрования с закрытым ключом, так как для шифрования и расшифровывания текста используется один и тот же примитивный многочлен и применяется однотипная операция XOR.

Для того, чтобы лучше понять принципы данного исследования в первую очередь необходимо знать, что такое шифр. *Шифр* - это совокупность заранее оговоренных способов преобразования исходного секретного сообщения с целью его защиты. Как говорилось выше, в рассмотрение берется две гипотезы, в первой из которой используется открытый текст - исходное сообщение, где символом является любой знак, в том числе буква, цифра или знак препинания. В данной работе будет использоваться английский алфавит, состоящий из 26 букв, а также знаки препинания: точка, запятая, восклицательный знак, вопросительный знак, кавычки. Данное дополнение обусловлено тем, что необходима минимальная избыточность работы программы, в связи, с чем была выбрана пяти битная кодировка, позволяющая верно зашифровать все 26 букв алфавита без учета регистра. Чтобы расшифрованный текст состоял из тех же символов, что и зашифрованный, необходимо дополнить исходный алфавит любыми символами, встречаемыми в исходном открытом тексте. Вообще, *алфавитом* в криптографии называется конечное множество используемых для кодирования информации символов, преобразование открытого текста в криптограмму - зашифрованием, а обратное действие - расшифрованием.

Еще одним важным определением в данной области является понятие *ключа* - информации, необходимой для шифрования и расшифрования сообщений. В данной работе система шифрования - любая система, использующаяся для обратимого изменения текста сообщения с целью сделать его непонятным для всех, кроме тех, кому оно предназначено - использует *регистр сдвига с линейно обратной связью (linear feedback shift register, LSFR*) на основе примитивного многочлена тридцать первой степени. Данный выбор обусловлен множественными преимуществами РСЛОС:

* высокое быстродействие создаваемых алгоритмов;
* пользование простейших битовых операций, которые аппаратно были реализованы практически во всех вычислительных устройствах;
* РСЛОС легко анализируются с использованием алгебраических методов;
* широкие исследования в области рекуррентных последовательностей с относительно удовлетворительными криптографическими свойствами;
* стойкие криптографические свойства.

Но наряду с этим, РСЛОС так же имеют недостатки (неэффективная программная реализация, линейность последовательности на выходе регистра позволяет однозначно определить многочлен обратной связи последовательным битам с помощью алгоритма Евклида, относительная легкость анализа алгебраическими методами увеличивает шансы на взлом), которыми в данной работе можно пренебречь.

# Часть 2. Изучение линейно – рекуррентных последовательностей

Прежде чем приступить к изучение рекуррентных последовательностей, как и в первой части данной главы, нам необходимо ввести некоторые определения.

Последовательностью над полем называют любую функцию : , заданную на множестве целых неотрицательных чисел и принимающую значения в поле.

*Последовательность* называют линейной рекуррентной последовательностью порядка надо полем , если существуют константы , такие, что

.

Важным фактом является то, что линейная рекуррентная последовательность реализуется описанной в данной работе схемой линейного регистра сдвига.

Принцип работы РСЛОС заключается в том, что значения, которые содержатся в ячейках накопления, умножаются на заданные коэффициенты и затем суммируются (в поле P), что приводит к следующему результату: происходит сдвиг информации в регистре, в пустую ячейку (крайнюю) записывается значение суммы, которые было вычислено.

Запишем теперь закон рекурсии, выражающий зависимость между знаками линейной рекуррентной последовательности:

.

В данном равенстве многочлен u будет называться *характеристическим многочленом ЛРП* (линейная рекуррентная последовательность), вектор – *начальный вектор ЛРП*. Если данный вектор имеет наименьшую степень, то он называется *минимальным многочленом ЛРП*, а сама степень в данном случае *линейной сложностью ЛРП* *u*, которая и определяет минимальную длину линейного регистра сдвига, реализующего данную последовательность.

Еще одним важным определением является определение периода последовательности: наименьшее натуральное число t, для которого такое, что для всех верно следующее равенство: называется *периодом последовательности*.

Если вернуться к выше описанному закону рекурсии, то можно увидеть, что если *P* является конечным полем из элементов, то тогда максимальное значение периода данной последовательности порядка *n* будет равно . Это очевидно, ведь нулевой начальный вектор создает последовательность из одних нулей, тогда как число различных заполнений регистра длины n равно .

Линейно рекуррентные последовательности, которые имеют максимально возможный период получили название *максимальных ЛРП*.

Чтобы определить значение периода ЛРП необходимо обратить внимание на свойства минимальных многочленов. Таким образом, для того, чтобы ЛРП порядка n над полем из pэлементов имела максимальный период, необходимо и достаточно, чтобы минимальный многочлен являлся *примитивным* – неприводимый многочлен, корень которого в мультипликативной группе поля разложения имеет порядок . Существует достаточное количество критериев проверки неприводимости многочлена над конечными полями и методов их построения, но доказательство примитивности неприводимого многочлена удается в редких случаях.

Задача построения примитивных многочленов не является тривиальной, решение которой даже в исключительных случаях связано со многими трудностями, отчего на практике используют таблицы неприводимых многочленов над кончеными полями.

*Утверждение 1.* Если – неприводимый многочлен над полем *GF(2)* ( обозначение конечного поля из определенного количества элементов) степени *n,* и - простое число, то – примитивный многочлен.

*Утверждение 2.* Неприводимый многочлен примитивен в том и только в том случае, когда для любого просто числа p, делящего , многочлен не сравним с 1 по модулю многочлена .

Для удобного вычисления очередного знака через предыдущие используется представление ЛРП в виде формулы общего члена последовательности, которая являет собой аналитическое выражение члена последовательности в виде функции от его номера.

Рассмотрим следующее предположение. Пусть и *Q –* поле из элементов, являющееся расширением поля *P.* Тогда функцией «след» из поля *Q* в поле *Р* называется отображение вида . В силу свойств конечных полей справедливы равенства:

которые означают, что искомая функция «след» является линейным отображением над полем *P*.

Утверждение 3. Пусть *F(x)* – примитивный многочлен степени n над полем *P* и *Q* – корень *F(x)* в поле *Q*. Тогда любая ненулевая мультиграмма встречается на периоде ЛРП и ровно раз, а число вхождений нулевой мультиграммы на единицу меньше.

Приведенные выше утверждения показывают, что ЛРП над полем позволяют обеспечить первые два из трех требований к псевдослучайным последовательностям, используемым при построении управляющих блоков поточных шифросистем. При выборе закона рекурсии можно гарантировать достаточную величину периода получаемой псевдослучайной последовательности, а также хорошие статистические качества. Действительно, ведь период последовательности совпадает с числом всех ненулевых векторов длиной равной степени минимального многочлена ЛРП, где каждый из них встречается из этих векторов на периоде последовательности ровно один раз.

При это стоит отметить, что аналитическое строение ЛРП довольно простое. Для определения начального вектора по некоторому отрезку последовательности достаточно решить несложную систему линейных уравнений. Поэтому при использовании РСЛОС в криптографических задачах следует учитывать процедуры, заметно повышающие сложность аналитического строения разрабатываемых ими ЛРП.

Часть 3. Изучение деревьев принятия решений и рандомизированных лесов

Прежде чем приступить к описанию алгоритма рандомизированного леса (от англ. Random - случайный), обратим свое внимание на понятие «Дерево принятия решений» (также могут встречаться понятия дерево классификации, регрессионное дерево) – средство поддержки принятия решений, которое используется в статистике и анализе данных при прогнозировании моделей. Структура такого дерева представляет собой «ветки» и «листья». На «ветках» записываются атрибуты, от которых зависит целевая функция, на «листьях» записываются значения целевой функции. Остальные узлы представляют собой атрибуты, отличающие друг от друга различные случаи. Для классификации нового случая необходимо спускаться вниз по дереву до «листа» и выдать соответствующее значение.

«Дерево решений» при анализе решений используется как визуальный и аналитический инструмент для поддержки принятия решений; так можно получит ожидаемые значения альтернатив. Как правило, «дерево решений» имеет три типа узла: узлы решения, вероятностные и замыкающие узлы.

Циклические элементы пример того, что не может содержать в себе дерево решений, то есть каждый новый лист может только расщепляться, в связи с чем, при построении дерева мы можем столкнуться с проблемой его размерности.

Существует два основных типа деревья решений:

* дерево для классификации (предсказываемый результат является классов, к которому принадлежат данные);
* дерево для регрессии (результат – вещественное число).

Основными преимуществами деревьев решений являются:

* простота реализации;
* наглядность;
* универсальность (задачи классификации и регрессии);
* возможность работы с неполным значением всех атрибутов данных;
* хорошая производительность в процессе классификации по уже построенному дереву.

Среди недостатков можно отметить нестабильность процесса, размерность дерева (является критическим фактором качества при решении задачи), неопределенность разделения на классы (особенно в сложных случаях). Говоря о более сложных алгоритмах, основанных на дереве решения, можно с уверенностью сказать, что они помогают избежать выше перечисленных недостатков.

Возвращаясь теперь к алгоритму машинного обучения «random forest», основанному на механизме использования ансамбля деревьев решений, опишем принципе его работы.

Данный алгоритм был предложен учеными Лео Брейманом и Адель Катлер. Алгоритм сочетает в себе пару основных идей: метод бэггинга Бреймана и метод случайных пространств, автором которого является Tim Kam Ho. Основными направлениями, где применяется данный метод являются задачи классификации, кластеризации и регрессии.

Опишем алгоритм обучения классификатора:

Пусть обучающая выборка состоит из N примеров, размерность пространства признаков равна М, а так же существует некий параметр m.

Все деревья ансамбля строятся независимо друг от друга по нижеприведенной схеме:

1. Сгенерируем случайную подвыборку с повторением размеров N из исходной обучающей выборки.
2. Построим обучающее дерево, классифицирующее примеры данной подвыборки. При создании дерева будем выбирать признак при создании очередного узла дерева, на основе которого производится разбиение, но не из всех M признаков, а только из m случайно выбранных.
3. Дерево строится до тех пор, пока не закончится подвыборка и она не подвергнется процедуру прунинга (отсечение ветвей).

Классификация объектов происходит путем голосования: каждое дерево ансамбля относит классифицируемый объект к одному из классов. Выигрывает класс, за который было отдано большое число деревьев.

Оптимальное количество деревьев подбирается таким бразом, чтоб возможно было минимизировать ошибку классификатора на тестовой выборке. При ее отсутствии минимизируется оценка ошибки out – of – bag: доля примеров обучающей выборки, неправильно классифицируемых ансамблем, если не учитывать голоса деревьев на примерах, входящих в их собственную обучающую выборку.

Основными достоинствами данного алгоритма является:

* способность эффективно обрабатывать данные с большим число признаков и классов;
* внутренняя способность модели к обобщению;
* нечувствительность к масштабированию;
* высокая параллелизируемость.

Естественным недостатком является большой размер итоговых моделей, что требует для K числа деревьев O(K) памяти для хранения модель.

Глава 2

# Часть 1. Реализация процедуры потокового шифрования с помощью линейно – рекуррентных последовательностей

Прежде чем приступить к детальному описанию реализации процедуры шифрования и дешифрования опишем общий алгоритм процесса:

* использование ключевого неприводимого многочлена для формирования линейной рекуррентной последовательности с помощью линейного регистра сдвига с обратной связью;
* генерация шифротекста на основное открытого текста с помощью выведенной последовательности;
* применение операции шифрования к шифротексту и как следствие вывод открытого текста (введенного изначально).

### 1.1 Генератор потока ключей

Потоковые шифры преобразуют открытый текст в шифротекст по одному биту за операцию. На рисунке ниже отображена простейшая реализация потокового шифра (*Рисунок 1*).



Рисунок 1. Процедура шифрования/ дешифрования

Данный поток ключей и поток битов открытого текста подвергаются операции XOR, называемой «исключающее или», результатом которой и будет служить поток битов широфтекста:

*.*

Операция дешифрования аналогична операции шифрования, но выполняется над битами шифротекста, используя тот же самый поток ключей, восстанавливая биты открытого текст:

*.*

Говоря о безопасности данного генератора, можно с уверенностью утверждать, что она полностью зависит от свойств генератора: выдавая поток ключей как бесконечную строку нулей, шифрованный текст полностью будет соответствовать открытому текста. В таком случае вся проведенная операция окажется бессмысленной. Если же генератор будет формировать поток ключей как бесконечный поток случайных битов, на выходе будет получена надежная система шифрования.

Рассмотри потоковый шифр, где генератор потока формирует псевдослучайный битовый поток, который частично или полностью похож на случайный, хотя в действительности детерминирован, и значит, может быть безошибочно воспроизведен при дешифровании. Получается, что чем ближе выход генератора потока ключей случайному, тем потребуется больше времени для взлома шифра, что, в принципе, очевидно. Если же наоборот, то используемая криптосистема будет легко подвержена взлому.

Опишем принцип работы генератора поток ключей. Сам генератор состоит из трех основных частей:

* внутреннее состояние описывает текущее состояние генератора потока ключей. Одинаковые потоки ключей (одинаковый ключ и одинаковое внутреннее состояние) формируют два генератора поток ключей;
* генерация потока ключей происходит путем использования функции выхода по внутреннему состоянию;
* генерация нового внутреннего состояния осуществляется с помощью функции следующего состояния.



Рисунок 2. Принцип работы генератора ключей

### 1.2 Принцип работы ЛРСОС.

Для того чтобы понять принцип работы разрабатываемого алгоритма, необходимо прежде всего понимать на чем он основан. Одним из основных компонентов алгоритма является регистр сдвига с линейной обратной связью, о котором уже неоднократно упоминалось выше.

Итак, РСЛОС - регистр сдвига битовых слов, у которого значение входного бита равно линейной булевой функции от значений остальных битов до сдвига; прежде всего, РСЛОС находит применение в криптографии для генерации псевдослучайных последовательностей битов.

Как правило, РСЛОС делится на два основных модуля:

* регистр сдвига;
* схема обратной связи, вычисляющая значение входного бита.

Первый модуль состоит из функциональных ячеек памяти. Одна ячейка памяти хранит в себе текущее состояние одного бита. Общее количество таких ячеек называют длиной регистра. Все биты нумеруются числами , где содержимое *i-*й ячейки обозначается через . Значение одного нового бита определяется до сдвига битов в регистре и только после сдвига записывается в ячейку 0, а из ячейки *L-1* извлекается очередной сгенерированный бит.

Функцией обратной связи для РЛОС является линейная булева функция от значений всех или некоторых битов регистра, которая выполняет умножение битов регистра на коэффициенты , где *i = 1, 2, ... L*. Количество коэффициентов совпадает с количеством битов регистра L. Коэффициенты принимают значения , при чем последний коэффициент , так как РСЛОС задается характеристическим многочленом степени *L*.

Наиболее употребляемой булевой линейной функцией является операция «XOR», которая используется и в данном алгоритме. При этом биты, являющиеся переменными функции обратной связи, называются отводами, а сам регистр называется конфигурацией Фибоначчи.

Управление регистром в аппаратных реализациях производится подачей сдвигающего импульса (иначе называемого [тактовым](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%BA%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) или синхроимпульсом) на все ячейки. Управление регистром в программных реализациях производится выполнением [цикла](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)). На каждой итерации цикла вычисляется функция обратной связи и выполняется [сдвиг](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B4%D0%B2%D0%B8%D0%B3) битов в слове.

Теперь непосредственно опишем алгоритм работы РСЛОС. В течение каждого такта РСЛОС выполняет следующие операции:

* чтение бита из ячейки L-1 - данный бит является очередным битов выходной последовательности;
* вычисление нового значения функцией обратной связи для ячейки 0 с использованием текущих значений ячеек;
* перемещение каждой i-й ячейки в следующую ячейку i+1 (при этом уже i = 0,1,...,L-2);
* в ячейку 0 записывается бит, который ранее был вычислен функцией обратной связи.

Приведем примеры формул, по которым могут быть вычислены значения

ячеек, если функцией обратной связи является операция «XOR»:

* ;
* .

РСЛОС обладает так же тремя основными свойствами:

* периодичность;
* линейная сложность;
* корреляционная независимость.

Чаще всего при программной реализации линейного регистра сдвига с обратной связью используют параллельную работу 16-ти регистров. В этой схеме используется массив слов, размер которого равен длине регистра сдвига, а каждый бит слова относится к своему регистру.

Рассмотрим регистр, который используется при реализации алгоритма в данной работе. Для него имеется отводная последовательность (31,30,29,27,25), что означает, что для генерации нового бита необходимо с помощью линейной операции XOR просуммировать 30-й, 29-й,28-й,26-й,24-й биты. Код для такого регистра будет приведен ниже, при описании принципа работы алгоритма.

Для более наглядного представления работы регистра сдвига с линейной обратной связью приведем пример генерируемой последовательности на основе готово характеристичного примитивного многочлена: . Исходя из приведенного многочлена, мы можем утверждать, что битами отвода будут 2-й и 0-й. Запишем формулу для функции обратной связи: . Допустим начальным состоянием регистра следующую последовательность . Приведем таблицу, в которой опишем следующие состояние регистра:

Таблица 1. Состояния регистра

|  |  |  |
| --- | --- | --- |
| Номер шага | Состояние | Генерируемый бит |
| 0 |  | - |
| 1 |  | 1 |
| 2 |  | 0 |
| 3 |  | 0 |
| 4 |  | 1 |
| 5 |  | 1 |
| 6 |  | 1 |
| 7 |  | 0 |

Поскольку внутреннее состояние на последнем приведенном шаге вернулось к начальному состоянию, то, начиная со следующего шага, произойдет повторение битов. То есть генерируемая последовательность выглядит следующим образом: . Следовательно, период данной последовательности равен 7, то есть максимально возможному значению. Такое событие происходит в силу примитивности изначально заданного многочлена.

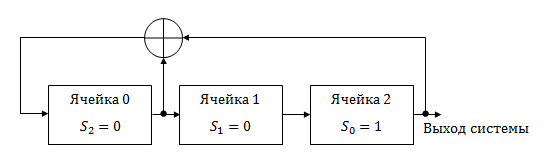


Рисунок 3. Пример работы РСЛОС

### 1.3 Критерий Вальда

В отличие от классического критерия Неймана -Пиросна последовательная процедура Вальда не предусматривает заранее фиксированного объема выборки *n* и решение о прекращении испытаний в зависимости от проведенных испытаний.

В критерии Вальда, являющемся обобщением классической процедуру, на каждой стадии эксперимента совокупность всевозможных выборок и объема разбивается на три непересекающихся множества . При попадании выборки в множество принимают гипотезу , испытания заканчивают; при попадании в множество принимают гипотезу и испытания заканчивают. В случае попадания выборки в область не принимают ни одной гипотезы, а производят следующее испытание и анализируют аналогично предыдущему выборку .

В процедуре Вальда объем выборки является случайной величиной. Поэтому за оптимальную процедуру построения областей естественно принять такую, которая при одних и тех же условиях имеет по сравнению с другими процедурами минимальное среднее значение , т.е.

Можно показать, что классическая процедура является частным случаем последовательной процедуры. Для этого достаточно выбрать следующее правило образования области:

,

где -заданный заранее объем выборки, - совокупность всевозможных выборок объема, 0 - пустое множество.

Приведенное выше соотношение означает, что решение может быть принято только на -ом шаге и классическая процедура является частным случаем процедуры Вальда с , где *v* - случайное число испытаний. Оказывается, что последовательная процедура может обеспечить при тех же самых , меньшее значение , чем соответствующее классическое число . Вальдом же была найдена оптимальная последовательная процедура определения областей , когда заданы вместо одного порога С (классическая процедура),два порога А и В.

Область задается неравенством . В этом случае принимается гипотеза и испытания прекращаются.

Область определяется неравенством . Если выборка удовлетворяет этому соотношению, то испытания продолжаются.

Если выборка удовлетворяет этому соотноешнию, то испытания продолжаются.

Область определяется соотношением. В этом случае принимается гипотеза и испытания прекращаются.

Функция определяется формулой:

Для однородной независимой выборки и двух значений параметра Вальдом была доказана оптимальность последовательной процедуры, обеспечивающей и найдены связи между параметрами .

Операционная характеристика в последовательной процедуре имеет вид:

*,*

где является корнем уравнения

.

Здесь *А* и *В -* два порога для коэффициента правдоподобия , причем в соответствии с определением имеем:

*.*

Данные соотношения можно пояснить следующим образом. По определению коэффициент правдоподобия равен отношение вероятностей того, что данная выборка соответствует гипотезе , т.е.

*.*

Согласно последовательному критерию вероятность выборки , приводящей к принятию гипотезы (и отклонению ), по крайней мере в *A* раз больше при гипотезе чем при

,

так же принимается при условии .

В отсутствии шумов разумным решением было бы принимать гипотезу каждый раз, когда , и гипотезу в обратном случае, т.к. выбор целесообразно делать в пользу той гипотезы, вероятность которой больше. В этом случае А=И=1. Однако, в случае наличия шумов или когда гипотезы перекрываются, получается зона неопределенности, следовательно, требуется увеличит *А* и уменьшить *В* по сравнению с единицей. Зона неопределенности будет определяться величиной *А-В*; данная зона в таком случае подлежит дальнейшему исследованию.

Аналогично классической процедуре для нормального закона при известной можно получить выражения для характеристик в явном виде. Последовательная процедура сводится к анализу величины с постоянным нижним и верхним порогами со случайным числом выборок. Перейдем к следующей величине для удобства: . Для этой величины нижний и верхний пороги будут равны соответственно: , где , , , так как .

Тогда, операционная характеристика может быть рассчитана по следующей формуле:

*.*

### 1.4 Описание работы алгоритма

Основной задачей в данной части главы является реализация процедуры шифрования и дешифрования с помощью ЛРСОС на основе рекуррентных последовательностей.

Шифрование текста производится с несколько этапов:

1. перевод текста в десятеричную кодировку по таблице ASCII,
2. формирование рекуррентной последовательности на основе выбранного примитивного многочлена для вывода ключа шифрования,
3. сложение векторов десятеричной кодировки символов теста и сформированного ключа; вывод суммы в качестве результата процедуры шифрования.

Для начала выберем примитивный многочлен из таблицы:

Приведенный выше многочлен используется в разрабатываемом алгоритме; в качестве многочлена может быть выбран любой многочлен, являющийся примитивным.

Напомним, что в качестве исходного алфавита используется английский алфавит, где количество букв является равным 26, в связи, с чем была выбрана пятибитная кодировка. Поэтому следующим шагом дополним исходный алфавит символами знаков препинания (что наиболее логично в рамках поставленной задачи), чтобы при обратной процедуре символы однозначно декодировались в исходные символы.

...//

(symbol >= 'a' && symbol <= 'z');

symbol -= 96; //1-26;

//...

Далее составим гистограмму частоты употребления знаков и биграмму употребления сочетаний символов в открытом тексте для определения верхнего и нижнего порога по критерию Вальда. Исходя из экспериментальных данных, можем утверждать, что верхним порогом будет являться число \_\_\_\_, а нижним - \_\_\_\_\_. Данные пороги определяют открытость текста, то есть является ли поданный на вход текст открытым или же, наоборот, шифрованным.

# Часть 2. Построение простого критерия на открытый текст на основе принципа запрещенных биграмм

# **Часть 3. Построение простого критерия на основе рандомизированного леса**

# **Глава 3**

Часть 1. Исследование эффективности критерия

Заключение и выводы

Список литературы

1. А.В. Бабаш, Г.П. Шанкин. Криптография. Аспекты защиты. Москва, Солон- пресс, 2007 г.
2. А.А. Харкевич. Борьба с помехами. Москва, Наука, 1965 г.
3. Злыгостев А.С. Информатика, Последовательный критерий Вальда. Москва, 2016.
4. Набебин А.А. Проблема оптимизации шифрования информации. Ученые записки государственного социального университета. Москва, №1/ 2010.
5. Воробьева Е.А., Лукьянова А.А.. Криптография, поточные шифры. Шифрование и дешифрование. Санкт – Петербург, Санкт – Петербургский государственный университет информационных технологий механики и оптики, 2005.
6. С.П. Чистяков. Случайные леса : обзор. Карелия, Институт прикаднх математических исследований Карельского научного центра Ран, №1, 2103.
7. С. Николаенко. Деревья принятия решений. Санкт – Петербург, ИТМО, 2006.