

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

|  |  |  |
| --- | --- | --- |
|  | | |
| Зав. кафедрой | | «КБИС» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | Д.А. Короченцев |
| (подпись) | |  |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |

**ОТЧЕТ**

по учебной, ознакомительной практике

в ФГАНУ НИИ «Спецвузавтоматика»

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. А. Котелевец

подпись, дата

Обозначение отчета УП.220000.000 Группа ВКБ11

Направление 10.05.01 Компьютерная безопасность

Направленность (профиль) Математические методы защиты информации

Руководитель практики:

от предприятия зам. директора по научной работе   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Ю. Гуфан

подпись, дата

от кафедры     ст. преподаватель                  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Э.Р. Типаева

подпись, дата

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись преподавателя

Ростов-на-Дону

2023 г



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

**ЗАДАНИЕ**

По учебной, ознакомительной практике

В ФГАНУ НИИ «Спецвузавтоматика»

в период с «5» июня 2023 г. по «1» июля 2023 г

Обучающийся Кирилл Александрович Котелевец

Обозначение отчета УП.220000.000 Группа ВКБ11

Срок представления отчета на кафедру «\_\_» \_\_\_\_\_\_\_\_ 2023 г.

Содержание индивидуального задания

Программная реализация шифров Виженера и «Двойной квадрат» Уитстона, а также изучение истории создания, принципы работы и их недостатки.

|  |  |  |
| --- | --- | --- |
| Руководитель практики от  кафедры | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Э. Р. Типаева |
| Руководитель практики от  предприятия | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | К. Ю. Гуфан |
| Задание принял к исполнению | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | К. А. Котелевец |



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника

Кафедра «Кибербезопасность информационных систем»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Зав. кафедрой | | | «КБИС» | |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | Д.А. Короченцев | |
| (подпись) | | |  | |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | | | |

**Рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| **№** | **Мероприятие** | **Срок выполнения** |
| 1 | Прохождение вводного и первичного инструктажа по охране труда на рабочем месте, и инструктажа по пожарной безопасности на объекте. | 02.06.2023 |
| 2 | Получение индивидуального задания. | 05.06.2023 |
| 3 | Ознакомление с теоретическим материалом. | 06.06.2023 |
| 4 | Разработка программы, реализующего шифр Виженера и Уитстона | 07.06.2023-16.06.2023 |
| 5 | Доработка программы | 17.06.2023 |
| 6 | Оформление отчёта по практике. | 18.06.2023-27.06.2023 |
| 7 | Защита отчёта по практике на предприятии. | 29.06.2023 |
| 8 | Защита отчёта по практике на кафедре. | 01.07.2023 |

Руководитель практики:

от предприятия зам. директора по научной работе   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Ю Гуфан

подпись, дата

от кафедры старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Э.Р. Типаева

подпись, дата

Ростов-на-Дону

2023

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

В данном разделе ежедневно, кратко и четко записываются выполняемые работы, и в конце каждой недели журнал представляется для проверки руководителю (от предприятия и университета) практики. При выполнении одной и той же работы несколько дней, в графе «дата» сделать запись «с \_\_\_по\_\_\_».

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Место работы | Выполняемые работы | Оценка руководителя |
| 02.06.2023 | ДГТУ | Прохождение вводного и  первичного инструктажа по охране  труда на рабочем месте, и  инструктажа по пожарной  безопасности на объекте. |  |
| 05.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Получение индивидуального  задания. |  |
| 06.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Изучение теоретического материала, касающегося шифра Виженера и шифра Уитстона |  |
| 07.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Настройки программы PyCharm для разработки программы, создание проекта |  |
| 08.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Изучение библиотек Python для программной реализации |  |
| 09.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Разработка основной логики алгоритма шифров. |  |
| С 10.06.2023 по 12.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Разработка архитектуры и алгоритмической схемы программы |  |
| С 13.06.2023 по 16.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Доработка алгоритмов шифрования и дешифрования |  |
| 17.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Отладка программы |  |
| С 18.06.2023 по 27.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Оформление отчёта по практике. |  |
| 29.06.2023 | ФГАНУ НИИ «Спецвузавтоматика» | Защита отчёта по практике на предприятии. |  |
| 01.07.2023 | Кафедра «КБИС» | Защита отчёта по практике на кафедре. |  |

от предприятия  зам. директора по научной работе \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Ю. Гуфан

подпись, дата

от кафедры   ст. преподаватель   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Э.Р. Типаева

подпись, дата

Содержание

[Введение 6](#_Toc138704712)

[1 Классический шифр Виженера 8](#_Toc138704713)

[1.2 История 8](#_Toc138704714)

[1.2 Описание 9](#_Toc138704715)

[1.3 Модификация шифра Виженера 11](#_Toc138704716)

[1.4 Критическая оценка криптостойкости шифра Виженера 13](#_Toc138704717)

[2 Шифр «Двойной квадрат» Уитсона 15](#_Toc138704718)

[2.1 Описание 16](#_Toc138704719)

[2.2 Алгоритм 17](#_Toc138704720)

[2.3 Критическая оценка криптостойкости шифра Уитстона 18](#_Toc138704721)

[3 Программная реализация 20](#_Toc138704722)

[3.1 Высокоуровневый язык программирования Python 20](#_Toc138704723)

[3.2. Программная реализация шифра Виженера 21](#_Toc138704724)

[3.3 Программная реализация шифра Уитстона 24](#_Toc138704725)

[3.4 Модуль verification 28](#_Toc138704726)

[3.5 Главный исполняемый файл 29](#_Toc138704727)

[Заключение 31](#_Toc138704728)

[Перечень использованных информационных ресурсов 32](#_Toc138704729)

[ПРИЛОЖЕНИЕ А 33](#_Toc138704730)

[ПРИЛОЖЕНИЕ Б 37](#_Toc138704731)

# Введение

Своим существованием Федеральное государственное автономное научное учреждение «Спецвузавтоматика» обязано инициативному изобретателю Алексею Аграновскому, выпускнику физического факультета Ростовского государственного университета, который с 1975 года работал в университетском вычислительном центре, начав с техника и закончив в должности начальника отдела автоматизации научных исследований. В 1992 году он основал на университетской базе конструкторское бюро «Спецвузавтоматика», которое в 2002 году превратилось во ФГАНУ НИИ «Специализированные вычислительные устройства защиты и автоматика» при Минобрнауки РФ. Аграновский оставался директором института до своей смерти в 2010 году.

Целью деятельности ФГАНУ НИИ «Спецвузавтоматика» является   
создание и внедрение информационных технологий для развития инновационного потенциала различных государственных структур по решению их собственных задач в области науки, техники и образования.

Сфера компетенции ФГАНУ НИИ «Спецвузавтоматика» достаточно обширна, так как она включает в себя все аспекты работы в сферах   
безопасности и защиты информации, управления проектами и внедрения инновационных технологий. Также институтом проводятся различные научные исследования в области телекоммуникаций, создания технических средств  
связи, защиты и обработки информации, разрабатывается программное обеспечение по заказу и создаются электронные устройства, начиная от идеи и заканчивая серийными образцами.

В Институте создана организационная структура, позволяющая развивать прикладные исследования и научные школы, осваивать новые технологии, реализовывать инновационные продукты.

Существует несколько подразделений:

1. Научно-исследовательские лаборатории (НИЛ);
2. Южный региональный аттестационный центр (ЮРАЦ);
3. Центр трансфера технологий и коммерциализации (ЦТТК);
4. Центр космических технологий «Арктурус» (ЦКТ);
5. Национальный центр противодействия терроризму и экстремизму в образовательной среде и сети Интернет (НЦПТИ);

И кроме всего прочего существуют такие приоритетные направления деятельности:

1. Выполнение работ и исследований в сфере информационных технологий, естественных и технических наук, информационной безопасности, инновационным методам разработки программного обеспечения, проблематике информационного противодействия терроризму и экстремизму в сети Интернет и образовательной среде, телекоммуникациям и управлению, созданию новых образовательных технологий;
2. Координация научных исследований по перспективным направлениям  
   в области безопасности Российской Федерации;
3. Разработка и внедрение в сферу образования перспективных системных и обучающих технологий;
4. Организация инновационной деятельности с учетом основных направлений социально-экономического развития Российской Федерации, реализации приоритетных направлений государственной политики Российской Федерации в научно-технической сфере;
5. Внедрение в производство новых технологий и техники по заказам заинтересованных предприятий и организаций;
6. Содействие развитию кадрового потенциала региона, в том числе  
   в инновационной и проектной деятельности.

## 1 Классический шифр Виженера

Шифр Виженера – это метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Данный метод является простой формой полиалфавитной замены. Впервые данный метод описывал Джовани Баттиста Белласо в 1553 году, однако уже в XIX веке получил имя Блеза Виженера, французского дипломата. Шифр Виженера очень прост для понимания и реализации, но является недоступным для простых методов криптоанализа.

### 1.2 История

В 1466 году Леон Альберти, знаменитый архитектор и философ, представил трактат о шифрах в папскую канцелярию. В его трактате рассматриваются разнообразные способы шифрования, в том числе маскировка открытого текста в некотором вспомогательном тексте. Он разработал шифр под названием «шифр, достойный королей». Это был многоалфавитный шифр, реализованный в виде шифровального диска. Альберти сделал большой прорыв в сфере криптографии и своими изобретениями значительно опередил свое время.

В 1518 году был сделан новый шаг благодаря появлению в Германии первой печатной книги по криптографии – Аббат Иоганн Тритемий, настоятель монастыря в Вюрцбурге, написал книгу «Полиграфия», в которой дается описание ряда шифров. Один из них использует «таблицу Тритемия» (ныне уже называется «таблицей Виженера») и развивает идею многоалфавитной замены. Однако в шифре Тритемия отсутствует ключ, секретом шифра является сам способ шифрования: первая буква исходного текста шифруется по первой строке, вторая по второй и так далее. После использования последней строки  
следующая буква вновь шифруется по первой строке. Следующий шаг в развитии предложенного Тритемием способа шифрования был сделан  
Джовани Белазо, который в 1553 году выпускает в свет свою брошюру  
«*Шифр синьора Белазо*». В этом шифре был ключ, являющийся паролем (фраза или слово).

Через какое-то время идеи Тритемия и Белазо развил Джовани Батиста Порта. Он предложил отказаться от алфавитного порядка следования букв в первой строке таблице Тритемия и заменить этот порядок на некоторый произвольный, являющийся ключом шифра. В своей книге «*О тайной переписке*» Порта предложил биграммный шифр, а также привел описание механического дискового устройства, реализующего биаграммную замену.

Посол Франции в Риме Блез де Виженер (Рис.1), познакомившись с трудами Тритемия Белазо, Кардано, Порта и Альберти, также увлекся криптографией. В 1585 году им был написан «*Трактат о шифрах*», в котором излагаются основы криптографии. Виженер объединил подходы Тритемия, Беллазо, Порта и осуществил свой шифр.



Рисунок 1 – Портрет «Блез Виженер»

### 1.2 Описание

В шифре Цезаря каждая буква алфавита сдвигается на несколько позиций; например, в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря  
с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая квадратом (таблицей) Виженера (Рис.2). Применительно к латинскому алфавиту таблица Виженера составляется из строк  
по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

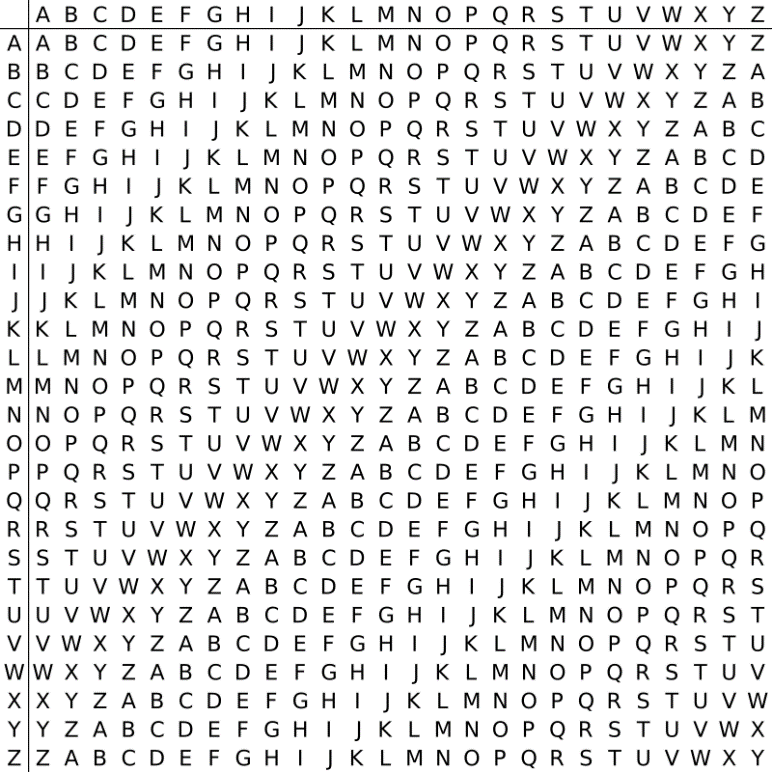


Рисунок 2 – Квадрат (таблица) Виженера

Например, предположим, что исходный текст имеет вид: «*BREADANDSALT*». Человек, посылающий сообщение, записывает ключевое слово «*GUEST*» циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: «*GUESTGUESTGU*». Первый символ теста B зашифрован последовательностью G, которая является первым символом ключа. Первый символ H шифрованного текста находится на пересечении строки G и столбца A в таблице Виженера. Остальная часть шифруется подобным способ и получается такой текст: «*HLISWGHHKTRN*»

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова;  
в данной строке находим первый символ зашифрованного текста. Столбец,  
в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

### 1.3 Модификация шифра Виженера

Шифр Виженера является одной из самых известных криптосистем, поэтому модификаций к шифру Виженера много, но среди них выделяются следующие: шифр Бофора, шифр Бегающего ключа, шифр Гронсфельда.

Шифр Бофора (Бьюфорта) назван именем адмирала Френсиса Бофора, который был изобретателем шкалы для определения скорости ветра. Шифр Бофора представляет из себя шифр Виженера, за одним исключением – в шифре Бофора строки квадрата Виженера записаны в обратном порядке.   
На рисунке 3 представлен Квадрат Бофора, использующийся при шифровании текста.



Рисунок 3 – Квадрат Бофора

После взлома шифра Виженера многие работали над его улучшением. Одним из главных недостатков шифра Виженера является повторение ключа. Это заметили и Вернам, который в процессе работы с шифром Виженера изобрел шифр Гилберта Вернама, и Чарльз Бэбидж, и Фридрих Вильгельм Касиски. Они решили заменить повторяющийся ключ, на ключ длинной равной исходному тексту. Проблема Бегущего ключа и шифра Виженера состоит в том,  
что взломщик имеет статистическую информацию о ключе. Если подобрать случайный ключ, и его длина будет равна длине сообщения, а сам он будет использоваться единожды, то теоретически шифр Виженера будет невозможно взломать.

Граф Гронсфельд создал похожую на шифр Виженера криптосистему, которая использовала всего 10 алфавитов вместо 26 в латинском алфавите. Преимущество такого шифра состояло в том, что в качестве ключевого слова он использовал последовательность целых чисел, повторяющихся до тех пор, пока ее длина не окажется равной длине исходного сообщения. Алгоритм шифрования схож с алгоритмом шифрования Виженера. Точно так же составляется таблица, содержащая алфавиты, при этом строки пронумерованы от 0 до 9  
и для символов каждой строки применяется сдвиг, каждая n-ая строка содержит алфавит n-1-ой строки со сдвигом на k позиций. Такой шифр был широко известен в Европе. На рисунке 4 представлена таблица Гронсфельда.

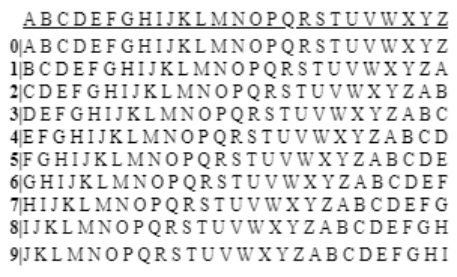


Рисунок 4 – Таблица Гросфельда

### 1.4 Критическая оценка криптостойкости шифра Виженера

Шифр Виженера, в силу многократно повторяющегося ключа, нельзя назвать абсолютно стойким к взлому. Это давно заметили многие криптоаналитики. Еще одним недостатком такой криптосистемы является то, что символы исходного текста просто заменяются на символы некоторого алфавита. Все символы, из которых состоят общеизвестные алфавиты, употребляются  
в письменной и устной речи с различной частотой. Например, для сбора данных о частоте употребления символов латинского алфавита, были анализированы произведения Байрона. Таким образом, зная длину ключевого слова можно приблизиться к нахождению исходного текста. Впервые такой метод анализа зашифрованных сообщений предложил арабский философ Аль Кинди.  
В 1863 году Фридрих Касиски публикует результаты своей работы, в которой  
он применил метод частотного анализа к зашифрованному тексту. В этой работе он описал алгоритм взлома шифра Виженера и показал, что данная криптосистема не безопасна. Однако в 1854 году Чарльз Бэббидж Совершил  
это же открытие, но публиковать свою работу он не стал. В 1920 году  
У. Фридман опубликовал результаты своих исследований, направленных  
на обнаружение исходного текста из шифротекста, зашифрованного алгоритмом Виженера. Фридман ввел новую величину, которая называется  
«Индексом совпадений». С его помощью, можно найти период гаммы наложения, по зашифрованному сообщению, полученному с помощью шифра Виженера.

При взломе шифра Виженера, необходимо знать длину  
ключа. Для того, чтобы найти длину ключевого слова, используется индекс совпадений. ИС является вероятностью того, что две взятые буквы  
в осмысленном тексте окажутся одинаковыми. В шифре Виженера  
минимальная длина ключевого слова k = 2, иначе наш шифр будет являться Сдвигом Цезаря.

Пусть I - конечный алфавит

Индекс совпадения будет выражаться следующей формулой:

fi - частота появления i-того символа в тексте J = i1,i2,…,iN

m - мощность алфавита

Предположительно, что длина ключа равна n, при n = 2.

Индекс совпадения для двух букв русского алфавита равен 0.0552  
(ИС = 0.0552) для латинского ИС = от 0.062 до 0.067

Из шифротекста выбирается каждая вторая буква, начиная с первой.

Считается длина текста N, далее пользуясь алфавитом, рассчитывается индекс совпадений для каждого символа. Допустим А встречается в тексте  
17 раз, а длина текста N = 402, тогда используя формулу нахождения  
ИС = 17\*16/402/401 = 0.00169, далее находится индекс совпадений  
для всех символов алфавита, и они складывается. Если Σ И Сi ≥ 0.0552  
для русского, то длина ключа была найдена верно. Иначе продолжается поиск длины ключа.

На втором шаге необходимо найти ключ. Зная его длину, разбивается  
текст на k групп. В каждой группе находим разность индексов  
наиболеечасто встречаемым символом и индексом «О» в русском алфавите,   
в латинском «E». Разность индексов — это индекс i – ого  
символ ключа применимого при шифровании. Результаты записываются в том порядке, в котором проиндексированы группы и записывается символ,  
находящиеся в русском алфавите под этими индексами. Получается приблизительный образ ключа. Приблизительный, потому что при большой длине ключа его частота в тексте будет меньше, а значит его будет  
труднее определить. Предполагая, что ключ был осмысленный, дальнейшее  
определение ключа возлагается на пользователя. Дешифруется  
шифротекст. Таким образом, для нахождения длины ключа  
необходимо рассмотреть вероятность появления всех символов в тексте.

## 2 Шифр «Двойной квадрат» Уитсона

В истории криптографии условно можно выделить четыре этапа: наивный, формальный, научный; компьютерный.

Для наивной криптографии (до начала XVI в.) характерно использование любых, обычно примитивных, способов запутывания противника относительно содержания шифруемых текстов. На начальном этапе для защиты информации использовались методы кодирования и стеганографии, которые родственны,  
но не тождественны криптографии.

Большинство из используемых шифров сводились к перестановке  
или моноалфавитной подстановке. Одним из первых зафиксированных примеров является шифр Цезаря, состоящий в замене каждой буквы исходного текста  
на другую, отстоящую от нее в алфавите на определенное число позиций. Другой шифр, полибианский квадрат, авторство которого приписывается греческому писателю Полибию, является общей моноалфавитной подстановкой, которая проводится с помощью случайно заполненной алфавитом квадратной таблицей  
(для греческого алфавита размер составляет 5 × 5). Каждая буква исходного текста заменяется на букву, стоящую в квадрате снизу от нее.

Этап формальной криптографии (конец XV – начало XX вв.) связан с появлением формализованных и относительно стойких  
к ручному криптоанализу шифров. В европейских странах это произошло  
в эпоху Возрождения, когда развитие науки и торговли вызвало  
спрос на надежные способы защиты информации. Важная роль на этом этапе принадлежит Леону Батисте Альберти, итальянскому архитектору,  
который одним из первых предложил многоалфавитную подстановку.  
Данный шифр, получивший имя дипломата XVI в. Блеза Вижинера,  
состоял в последовательном «сложении» букв исходного текста  
с ключом (процедуру можно облегчить с помощью специальной таблицы). Его работа «Трактат о шифре» (1466 г.) считается первой научной работой по криптологии.

Одной из первых печатных работ, в которой обобщены и сформулированы известные на тот момент алгоритмы шифрования, является труд  
«Полиграфия» (1508 г.) немецкого аббата Иоганна Трисемуса. Ему принадлежат два небольших, но важных открытия: способ заполнения полибианского квадрата  
(первые позиции заполняются с помощью легко запоминаемого ключевого слова,  
остальные – оставшимися буквами алфавита) и шифрование пар букв (биграмм).

Простым, но стойким способом многоалфавитной замены (подстановки биграмм) является шифр Плейфера, который был открыт в начале XIX в. Чарльзом Уитстоном. Уитстону принадлежит и важное усовершенствование – шифрование «двойным квадратом». Шифры Плейфера и Уитстона использовались вплоть до первой мировой войны, так как с трудом поддавались ручному криптоанализу.

### 2.1 Описание

Шифр Уитстона или Шифр двойного квадрата – это ручная симметрическая техника шифрования, изобретенная английским физиком Чарльзом Уитстоном в 1854 году. Данная система является улучшенной версией шифра Плейфера, в котором аналогичным образом используется замена биграмм, но только с помощью одного квадрата. Однако в тоже время шифр является упрощенной версией шифра четырех квадратов, изобретенного известным французским криптографом Феликсом Деластелем в 1902 году.

Данный метод шифрует пары букв и попадает в категорию шифров, известных как полиграфические подстановочные шифры, который действуют  
на отдельные символы.

Для шифрования используются две матрицы 5x5, но можно делать произвольные и не обязательно квадратные размеры. Данные матрицы находятся либо одна под другой, либо друг напротив друга. Каждая из них в первую  
очередь заполняется соответствующим ключевым словом, не записывая повторяющиеся буквы, затем в оставшиеся ячейки матрицы записываются  
по порядку символы алфавита, которые не были ранее использованы. Ключевое слово может быть записано двумя способами:

1. В верхней строке матрице слева направо;
2. По спирали из левого верхнего угла к центру;

Таким образом и заполняется матрица. Также есть особенность, что ключ шифра Уитстона обычно состоит из двух слов (по одному слову на каждую матрицу). На рисунке 5 представлен вертикальный шифр Уитстона с ключевыми словами «example» и «keyword»

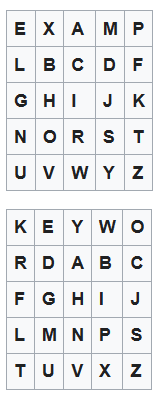


Рисунок 5 – пример вертикального шифра Уитстона

### 2.2 Алгоритм

Система шифрования с помощью двух квадратов схожа с системой, используемой в шифре четырех квадратов, но открытый и зашифрованный тексты используют одинаковые матрицы.

Чтобы зашифровать сообщение, необходимо выполнить такой алгоритм:

1. Разбить сообщение на биграммы;
2. Выбираем необходимый способ шифрования:
   1. В вертикальном способе первый символ биграммы находим в верхней матрице, второй в нижней;
   2. В горизонтальном способе первый символ находим в левой матрице, второй – в правой;
3. Определяем положения углов получившегося прямоугольника относительно друг друга. В случае, если буквы исходной биграммы сообщения находятся в одной строке (горизонтальный способ), то первую букву зашифрованной биграммы берем из левой матрицы в том по счету столбце, в каком находится вторая буква исходной биграммы. Вторая же буква шифрованной биграммы берется из второй матрицы в столбце, по счету которого находится первая буква исходной биграммы сообщения. Аналогично в способе вертикального шифрования.

### Критическая оценка криптостойкости шифра Уитстона

Как и большинство шифров формальной криптографии, шифр Уитстона также может быть легко взломан, если имеется достаточный объём текста. Получение ключа является относительно простым, если известны шифрованный и обычный текст. Когда известен только зашифрованный текст, криптоаналитики анализируют соответствие между частотой появления биграмм в зашифрованном тексте и известной частотой появления биграмм в языке, на котором написано сообщение.

Шифр Уитстона подобен шифру Плейфера с использованием  
для шифрования текста только одного квадрата, что упрощает идентификацию текста. Однако система шифрования Плейфера имеет ряд недостатковперед шифром Уитстона. Шифр Плейфера достаточно легко поддается криптоанализу *Random-restart hill climbing*. Также, другая отличительная особенность состоит в том, что в шифротексте Плейфера не встречаются биграммы с повторяющимися символами. Поэтому если в достаточно большом зашифрованном тексте отсутствуют биграммы с повторяющимися символами, то с большой долей вероятности можно утверждать, что исходный текст был закодирован с помощью шифра Плейфера.

Так как шифр Плейфера был взломан ещё во времена Первой мировой войны, шифр двойного квадрата активно использовался во Второй мировой войне немецкой армией, ВВС и полицией. Но и этот шифр был взломан в Блечли-парк, потому что немцы использовали один и тот же шаблон сообщения. В восьми сообщениях, зашифрованных двойным квадратом, были использованы цифры от одного до двенадцати, это и дало возможность достаточно легко взломать его.

## 3 Программная реализация

Шифр Виженера и Уитстона будут реализованы на языке Python, среда разработки – IDE PyCharm. Проект будет состоять из 4 файлов:

1. «Main.py» – главный исполняемый файл, в котором будет реализован выбор между шифром Виженера и Уистона;
2. «Vijener.py» – шифр Виженера;
3. «Wheatstone.py» – шифр Уитстона;
4. «Verification.py» – файл, в котором реализованы функции для проверки на правильность ввода.

### Высокоуровневый язык программирования Python

Python является высокоуровневым языком программирования общего назначения и ориентирован на лаконичность и простоту кода, а также на легкость его чтения. Синтаксис ядра языка минималистичен, а стандартная библиотека языка Python включает большой объем различных функций и 21 методов. Помимо стандартной библиотеки для языка Python существует большое количество дополнительных библиотек, которые содержат огромный набор полезных инструментов, применимых к ряду задач. Язык обладает динамической типизацией, на что уже акцентировалось внимание, автоматическим управлением памятью, механизмом обработки исключений, поддержкой многопоточных вычислений и высокоуровневой структурой данных. Так же для языка Python существует множество фреймворков, которые помогают ускорить работу написания кода. Как уже говорилось, язык программирования Python ориентирован на простоту и читаемость написанного кода. Поэтому человеку, с небольшим опытом программирования или программирующему на другом языке программирования, не составит труда разобрать и понять код, написанный на Python’е. Этот язык хорошо подходит для удобной и лаконичной реализации программного продукта для рабочего стола под управлением ОС Windows.

### 3.2. Программная реализация шифра Виженера

Программная реализация шифра Виженера будет осуществлена на языке программирования Python версии 3.11. Данный алгоритм будет реализован  
как вызываемая функция в файле «main.py»

Для начала импортируется необходимая библиотека (рис 7).

from itertools import cycle – происходит импорт функции cycle из модуля itertools. Функция cycle возвращает бесконечный итератор, который повторяет элементы из переданной ему последовательности.

from verification import \* – происходит импорт всех функции из модуля verification, который содержит функции для проверки вводимых пользователем данных.

import string – импортируем модуль string, который содержит список символов ASCII.

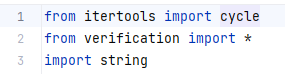


Рисунок 7 – импорт библиотек

Затем задается переменная ABC (рис. 8), содержащая в себе все буквы латинского алфавита и пробел, которые могут быть зашифрованы.



Рисунок 8 – алфавит

На рисунке 9 определяется функцию encode, которая принимает на вход текст и ключ, и возвращает зашифрованный текст, и функцию decode, которая принимает на вход зашифрованный текст и ключ, и возвращает расшифрованный текст. В 10 строке кода реализуется лямбда-функцию FV, которая принимает кортеж из двух символов и возвращает символ шифротекста, соответствующий позиции полученной с помощью операции сложения индексов символов  
из переданных ей элементов. % len(ABC) используется для обеспечения цикличности алфавита. В 11 строке кода применяется функция FV к каждому кортежу, состоящему из символов исходного текста и символа ключа, используя функции map и zip. Полученные символы шифротекста объединяются в строку  
с помощью метода join. На 14 строке определяется лямбда-функция FV, которая принимает кортеж из двух символов и возвращает символ исходного текста, соответствующий позиции полученной с помощью операции вычитания индекса символа ключа из индекса символа шифротекста. % len(ABC) используется для обеспечения цикличности алфавита. На 15 строке применяется функция FV  
к каждому кортежу, состоящему из символов зашифрованного текста и символа ключа, используя функции map и zip. Полученные символы расшифрованного текста объединяются в строку с помощью метода join.

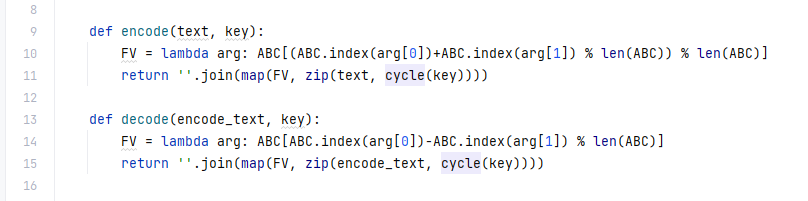


Рисунок 9 – Функции шифрования и расшифрования

На рисунке 10 реализован код, который позволяет ввести текст  
с клавиатуры с проверкой и получить необходимый нам результат.

Mode = verification\_Mode() – вызывается функция verification\_Mode  
из модуля verification для проверки и выбора режима работы программы (шифрование или расшифрование) и сохраняется результат в переменную Mode.

text = verification\_ABC(ABC) – вызывается функция verification\_ABC из модуля verification для проверки текста, который нужно зашифровать или расшифровать, и сохраняется результат в переменную text.

key = verification\_ABC(ABC) – вызывается функция verification\_ABC из модуля verification для проверки ключа и сохраняется результат в переменную key. Далее в 25 строчке кода в соответствии с выбранным пользователем выводится необходимый нам результат на экран. На рисунках 11 и 12 показан пример работы кода.

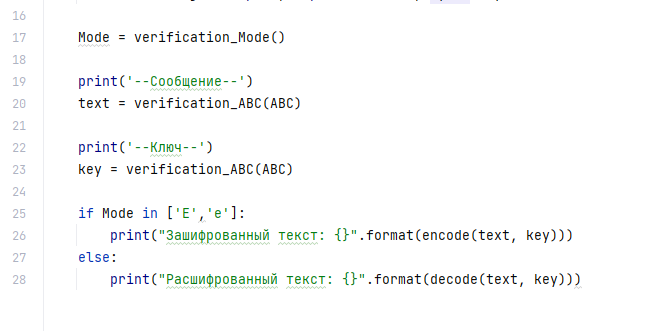


Рисунок 10 – Часть кода программы для реализации шифра Виженера

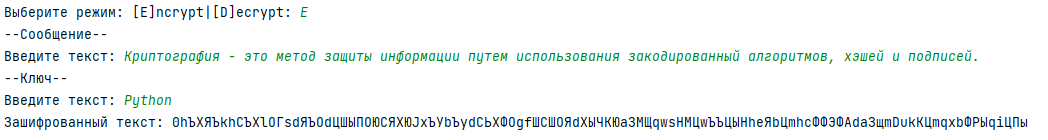


Рисунок 11 – Вывод программой зашифрованного сообщения

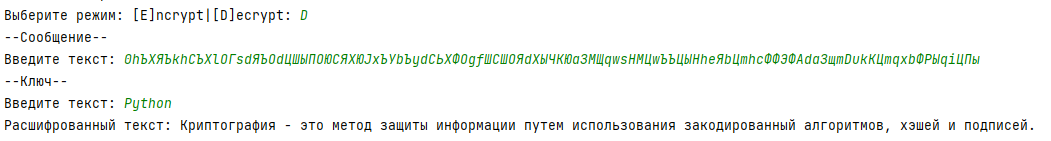


Рисунок 12 – Вывод программой расшифрованного сообщения

Таким образом с помощью написанного мною кода на языке программирования Python можно осуществить алгоритм шифрования Виженера.

### 3.3 Программная реализация шифра Уитстона

На рисунке 13 представлена строка, импортирующая необходимые библиотеки. import string - импорт модуля string, который содержит строковые константы для разных типов символов (буквы, цифры, знаки препинания и т.д.).   
from verification import \* - импорт всех функций из модуля verification. Данный алгоритм также будет реализован как вызываемая функция в файле «main.py».

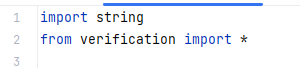


Рисунок 13 – импорт модулей

Далее идет объявление переменной ABC, которая является строкой, состоящей из русских и латинских букв, цифр, знаков препинания и пробела. (рис. 14)

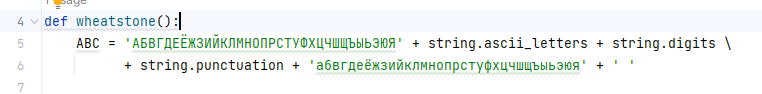


Рисунок 14 – Задание переменной ABC

Далее задается функция (рис. 15), которая создает двойную квадратную матрицу из заданного ключа. Сначала она преобразует ключ в нижний регистр и удаляет любые пробелы. Затем инициализируется пустой список с именем square. Она перебирает каждый символ в ключе и проверяет, что символ еще не находится в списке square и символ присутствует в допустимых символах, содержащихся в строке ABC. Если оба условия выполнены, то она добавляет символ в список square. Затем она перебирает каждый символ в ABC и добавляет любой символ, который еще не находится в списке square. Наконец, функция возвращает список square, в котором будут содержаться все символы, необходимые для двойной квадратной матрицы.

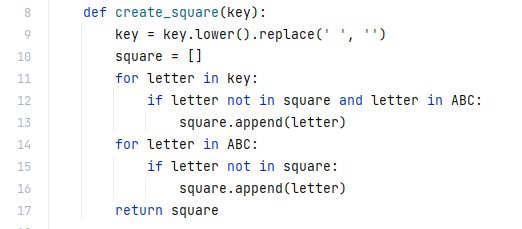


Рисунок 15 – Задание функции create\_square

Далее реализована функция encode (рис. 16), которая принимает  
на вход обычный текст (plaintext) и два ключа (key1 и key2). Создается  
двойная квадратная матрица с использованием create\_square функции  
и хранится в переменных square1 и square2. Затем, создается пустая  
строка-шифротекст (ciphertext). plaintext приводится к нижнему регистру, удаляются пробелы и переносы строк. Если длина plaintext нечетная,  
то к plaintext добавляется пробел. Затем она перебирает plaintext по два  
символа за раз. Для каждой пары символов определяется позиция  
(индексы строки и столбца) каждого символа в его соответствующей матрице. Затем извлекается символ на пересечении этих двух позиций, обращаясь  
к элементам square1 и square2. Наконец, эта символьная пара добавляется  
к переменной ciphertext.

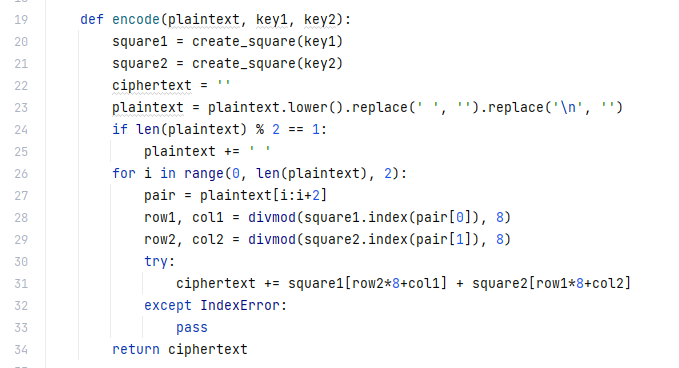


Рисунок 16 – Задание функции encode

Также реализована и функция decode (рис. 17), которая дешифрует зашифрованный текст (ciphertext) с использованием двух ключей (key1 и key2). Создается двойная квадратная матрица с использованием create\_square функции  
и хранится в переменных square1 и square2. Затем, создается пустая строка-расшифрованный текст (plaintext). Для каждой пары символов в ciphertext определяется позиция (индексы строки и столбца) каждого символа в его соответствующей матрице. Затем извлекается символ на пересечении этих двух позиций, обращаясь к элементам square1 и square2. Наконец, эта символьная пара добавляется к переменной plaintext.

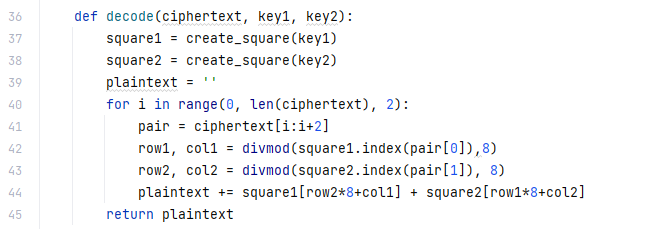


Рисунок 17 – Задание функции decode

На рисунке 18 представлены строки, которые запрашивают у пользователя два ключа и не зашифрованный текст с проверкой, извлекаемой из модуля verification. Затем, они шифруют plaintext с использованием encode функции и дешифруют его с помощью decode функции. Наконец, он выводит на экран исходный текст, зашифрованный текст и расшифрованный текст.



Рисунок 18 – Последняя часть кода

Общий результат - код использует метод двойной квадратной матрицы шифра Playfair для шифрования и дешифрования обычного текста. Разница заключается в том, что вместо использования ключевого слова для генерации одной квадратной матрицы, он использует два ключа для создания двух отдельных квадратных матриц. Эти матрицы используются для замены каждой пары символов в исходном тексте. Также есть особенность, что зашифрованный и расшифрованный текст не имеют пробелов. Это делает текст более компактным и труднее поддающимся анализу.

На рисунке 19 представлен пример работы шифра Уитстона.

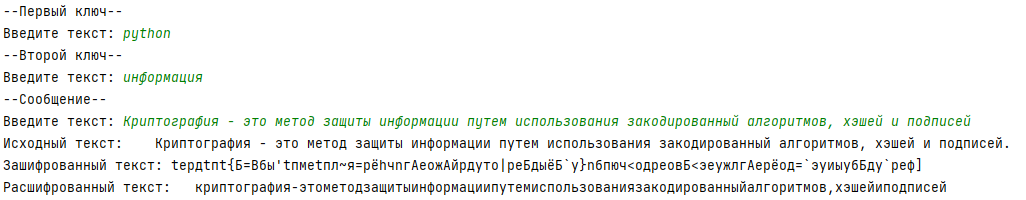


Рисунок 19 – Пример работы шифра Уитстона

### 3.4 Модуль verification

Модуль содержит три функции, каждая из которых выполняет проверку  
на правильность введённых пользователем данных для конкретной задачи.

Функция verification\_ABC(ABC) (рис. 20) принимает аргумент  
ABC, который является строкой символов, допустимых для ввода пользователем. Данная функция проверяет, что введенный пользователем текст содержит  
только символы из строки ABC. Если ввод недопустим, то выводится  
сообщение об ошибке и запускается рекурсия функции, чтобы  
попросить пользователя ввести данные еще раз.

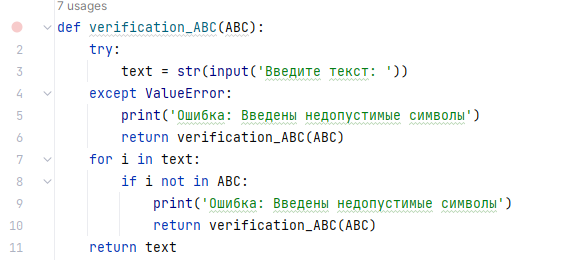


Рисунок 20 – Функция verification\_ABC(ABC)

Функция verification\_Mode() (рис. 21) не принимает аргументов.  
Она предназначена для проверки выбранного пользователем режима работы:   
шифрование или дешифрование. Данная функция просит пользователя  
ввести букву E или D в верхнем или нижнем регистре. Если ввод недопустим,  
то выводится сообщение об ошибке и запускается рекурсия функции,  
чтобы попросить пользователя ввести данные еще раз.



Рисунок 21 – Функция verification\_Mode()

Функция verification\_Main() также не принимает аргументов. Она предназначена для проверки выбранного пользователем номера от 0 до 2. Этот номер указывает на выбор конкретного алгоритма шифрования или дешифрования. Если ввод недопустим, то выводится сообщение об ошибке и запускается рекурсия функции, чтобы попросить пользователя ввести данные еще раз. Если же ввод правильный, то функция возвращает этот номер.

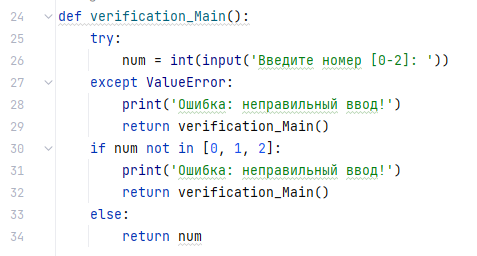


Рисунок 22 – Функция verification\_Main()

### 3.5 Главный исполняемый файл

Данный файл позволяет пользователю самостоятельно выбрать необходимый ему шифр. На рисунке 23 показан импорт необходимых модулей: «*wheatstone*», «*Vijener*» и «*verification*».

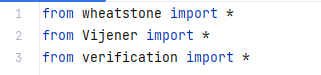


Рисунок 23 – Импорт модулей

На рисунке 24 показана часть кода, на которой представлен цикл, который позволяет непрерывно использовать программу, пока не будет выбрано  
число 0 – для окончания программы. В случае выбора цифры 1 запускается функция Vijener() из модуля Vijener, при цифре 2 – запускается аналогично шифр Уитстона.

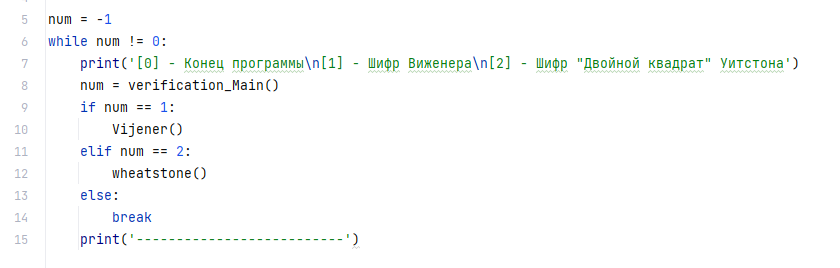


Рисунок 24 – Код реализующий выбор шифра

# Заключение

В процессе выполнения задания на учебную практику была поставлена задача реализовать программное средство, использующее алгоритмы шифрования Виженера и Уитстона.

Во время выполнения задания был изучен и реализован алгоритм шифрования Виженера, а также обнаружены модификации данного шифра. Так же был проведен анализ алгоритма шифрования Виженера и дана оценка его криптостойкости.

Изучение и реализация шифров Виженера и Уитстона позволили расширить знания в области алгоритмов и криптографии. Были освоены принципы работы шифров, изучены недостатки и достоинства, применение в различных задачах.

В заключение отчета по программной реализации шифра Виженера и Уитстона можно сказать, что оба метода представляют собой достаточно надежные способы шифрования текста. Шифр Виженера основывается на использовании ключевого слова, что делает его более безопасным, чем многие другие методы шифрования. Однако, он может быть взломан с помощью известных атак, таких как атака Касиски или метод грубой силы.

Шифр Уитстона, в свою очередь, является более простым и менее защищенным методом, поскольку его ключ состоит из чисел, которые могут быть легко угаданы. Однако, он может быть полезен для тех случаев, когда требуется быстрый и простой способ шифрования текста.

Несмотря на различия между этими двумя методами, оба они могут быть полезными инструментами для защиты конфиденциальной информации. Решение, какой метод лучше использовать, зависит от конкретной ситуации и уровня защиты, который необходим для данных, которые должны быть зашифрованы.

# **Перечень использованных информационных ресурсов**

1. Кирпичников А. П., Хайбуллина З. М. Криптографические методы защиты компьютерной информации: учебное пособие – Казань.: Изд-во КНИГУ,   
   2018 – 100 с. – ISBN 978-5-7882-2052-9.
2. Котов Ю. А. Криптографические методы защиты информации: шифры: учебное пособие — Новосибирск.: Изд-во НГТУ, 2018. — 59 с. —   
   ISBN 978-5-7782-2959-4.
3. Спицын В. Г. Информационная безопасность вычислительной техники: учебное пособие — Томск: Эль Контент, 2019. — 148 с. —   
   ISBN 978-5-4332-0020-3.
4. Википедия: Шифр Виженера [Электронный ресурс] URL: [Шифр Виженера — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%92%D0%B8%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0) (дата обращения 18.06.2023).
5. Википедия: Шифр Уитстона [Электронный ресурс] URL: [Шифр Уитстона — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A3%D0%B8%D1%82%D1%81%D1%82%D0%BE%D0%BD%D0%B0) (дата обращения 17.06.2023).

# ПРИЛОЖЕНИЕ А

**Техническое задание на программное средство**

|  |  |
| --- | --- |
| «СОГЛАСОВАННО» | «УТВЕРЖДЕНО» |
| Руководитель проекта | Зав. кафедрой «КБИС» |
| \_ / Э.Р. Типаева / | \_\_\_\_\_\_\_\_\_/ Д.А. Короченцев / |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023г. | «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023г. |

**А.1 Введение**

Наименование разрабатываемого программного средства: «Программная реализация систем шифрования Виженера и шифра «двойной квадрат» Уитстона»

Область применения:

* лабораторные работы по предмету алгоритмы и структуры данных;
* личное пользование приложением.
* защита текстовых данных

**А.2 Основания для разработки**

Разработка ведётся на основании документа «Положение о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования», утверждённого министерством образования и науки российской федерации 4.05.2018.

Предметным основанием является задание на учебную практику.

**А.3 Назначение разработки**

Функциональное назначение программного средства заключается в изучении шифрования Виженера и шифра «двойной квадрат» Уитстона для решения различных прикладных задач в программировании.

**А.4 Требование к программе или программному изделию**

**А.4.1 Требование к функциональным характеристикам**

Программа корректно использует алгоритм Виженера и Уитстона

Состав выполняемых функций:

* работать под управлением ОС Windows 7, Windows 8/8.1, а также Windows 10;
* шифрование и расшифровывание алгоритмом Виженера.
* шифрование и расшифровывание алгоритмом Уитстона

Входные данные:

* текстовое сообщение;
* текст-ключ;

Выходные данные:

* зашифрованный и расшифрованный текст;

Данные программой хранятся до следующего использование программы или выхода из нее.

**А.4.2 Требование надежности**

Разрабатываемая программа должна удовлетворять следующим требования к надежности: отказы программы вследствие внутренних ошибок не допустим;

**А.4.3 Условия эксплуатации**

Для функционирования программного продукта необходимо соблюдение всех требований и правил эксплуатации компьютерной техники. Программное средство обслуживается одним пользователем.

**А.4.4 Требование к составу и параметрам технических средств**

Состав технических средств:

* 10 Мб свободного места на диске; 32 или 64-разрядный процессор с частотой не ниже 1 ГГц;
* Монитор VGA или HDMI;
* Клавиатура и мышь;
* Оперативная память не меньше 256 Мб;
* Интерпретатор Python 3.11 требуется.

**А.4.5 Требования к информационно и программной совместимости**

Программное средство требует семейство операционных систем GNU/Linux для взаимодействия с пользователем через терминал.

**А.4.6 Требования к маркировке и упаковке**

Требования к маркировке не предъявляются. Требования к упаковке определяются требованиями к хранению носителей информации, содержащих ПО.

**А.4.7 Требования к транспортировке и хранению**

Требования к транспортировке и хранению определяются требованиями к транспортировке и хранению носителей информации, содержащих ПО.

**А.6 Стадии и этапы разработки**

Изучение предметной области (от 06.06.23 до 11.06.23);

Разбор технического задания (от 06.06.23 до 07.06.23);

Анализ работы алгоритма Виженера и Уитстона (06.06.23 до 09.06.23);

Разработка алгоритма Виженера и Уитстона (от 08.06.23 до 16.06.23);

Испытание программного средства (от 17.06.23 до 20.06.23).

**А.7 Порядок и контроль приемки**

Порядок и контроль приемки определяется руководством кафедры КБИС Донского государственного технического университета.

Разработчик технического задания: Котелевец Кирилл Александрович

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2023г \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

# ПРИЛОЖЕНИЕ Б

Листинг 1 – Реализация «Vijener.py»:

from itertools import cycle  
from verification import \*  
import string

def Vijener():  
 ABC = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz'  
  
  
 def encode(text, key):  
 FV = lambda arg: ABC[(ABC.index(arg[0])+ABC.index(arg[1]) % len(ABC)) % len(ABC)]  
 return ''.join(map(FV, zip(text, cycle(key))))  
  
 def decode(encode\_text, key):  
 FV = lambda arg: ABC[ABC.index(arg[0])-ABC.index(arg[1]) % len(ABC)]  
 return ''.join(map(FV, zip(encode\_text, cycle(key))))  
  
 Mode = verification\_Mode()  
  
 print('--Сообщение--')  
 text = verification\_ABC(ABC)  
  
 print('--Ключ--')  
 key = verification\_ABC(ABC)  
  
 if Mode in ['E','e']:  
 print("Зашифрованный текст: {}".format(encode(text, key)))  
 else:  
 print("Расшифрованный текст: {}".format(decode(text, key)))

Листинг 2 – Реализация «wheatstone.py»:

import string  
from verification import \*  
  
def wheatstone():  
 ABC = 'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ' + string.ascii\_letters + string.digits \  
 + string.punctuation + 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя' + ' '  
  
 def create\_square(key):  
 key = key.lower().replace(' ', '')  
 square = []  
 for letter in key:  
 if letter not in square and letter in ABC:  
 square.append(letter)  
 for letter in ABC:  
 if letter not in square:  
 square.append(letter)  
 return square  
  
 def encode(plaintext, key1, key2):  
 square1 = create\_square(key1)  
 square2 = create\_square(key2)  
 ciphertext = ''  
 plaintext = plaintext.lower().replace(' ', '').replace('\n', '')  
 if len(plaintext) % 2 == 1:  
 plaintext += ' '  
 for i in range(0, len(plaintext), 2):  
 pair = plaintext[i:i+2]  
 row1, col1 = divmod(square1.index(pair[0]), 8)  
 row2, col2 = divmod(square2.index(pair[1]), 8)  
 try:  
 ciphertext += square1[row2\*8+col1] + square2[row1\*8+col2]  
 except IndexError:  
 pass  
 return ciphertext  
  
 def decode(ciphertext, key1, key2):  
 square1 = create\_square(key1)  
 square2 = create\_square(key2)  
 plaintext = ''  
 for i in range(0, len(ciphertext), 2):  
 pair = ciphertext[i:i+2]  
 row1, col1 = divmod(square1.index(pair[0]),8)  
 row2, col2 = divmod(square2.index(pair[1]), 8)  
 plaintext += square1[row2\*8+col1] + square2[row1\*8+col2]  
 return plaintext  
  
 print('--Первый ключ--')  
 key1 = verification\_ABC(ABC)  
  
 print('--Второй ключ--')  
 key2 = verification\_ABC(ABC)  
  
 print('--Сообщение--')  
 plaintext = verification\_ABC(ABC) + '.'  
  
 ciphertext = encode(plaintext, key1, key2)  
 decoded\_plaintext = decode(ciphertext, key1, key2)  
  
 print(«Исходный текст: «, plaintext)  
 print(«Зашифрованный текст:», ciphertext)  
 print(«Расшифрованный текст: «, decoded\_plaintext)

Листинг 3– Реализация «verification.py»:

def verification\_ABC(ABC):  
 try:  
 text = str(input('Введите текст: '))  
 except ValueError:  
 print('Ошибка: Введены недопустимые символы')  
 return verification\_ABC(ABC)  
 for i in text:  
 if i not in ABC:  
 print('Ошибка: Введены недопустимые символы')  
 return verification\_ABC(ABC)  
 return text  
  
def verification\_Mode():  
 try:  
 mode = str(input('Выберите режим: [E]ncrypt|[D]ecrypt: ').upper())  
 except ValueError:  
 print('Ошибка, неправильный ввод!')  
 return verification\_Mode()  
 if mode not in ['E', 'D', 'e', 'd','Encrypt','encrypt','ENCRYPT','Decrypt','decrypt','DECRYPT','[D]','[d]','[E]','[e]']:  
 print('Ошибка, неправильный ввод!');  
 return verification\_Mode()  
 return mode  
  
def verification\_Main():  
 try:  
 num = int(input('Введите номер [0-2]: '))  
 except ValueError:  
 print('Ошибка: неправильный ввод!')  
 return verification\_Main()  
 if num not in [0, 1, 2]:  
 print('Ошибка: неправильный ввод!')  
 return verification\_Main()  
 else:  
 return num

Листинг 4– Реализация «main.py»:

from wheatstone import \*  
from Vijener import \*  
from verification import \*  
  
num = -1  
while num != 0:  
 print('[0] - Конец программы\n[1] - Шифр Виженера\n[2] - Шифр «Двойной квадрат» Уитстона')

num = verification\_Main()  
 if num == 1:  
 Vijener()  
 elif num == 2:  
 wheatstone()  
 else:  
 break  
 print('--------------------------')

ОТЗЫВ-ХАРАКТЕРИСТИКА

Обучающийся Котелевец Кирилл Александрович

группа ВКБ11

кафедра «Кибербезопасность информационных систем»

вид практики учебная практика

наименование места практики ФГАНУ НИИ «Спецвузавтоматика»

(наименование предприятия, структурного подразделения)

Обучающийся выполнил задания программы практики

Программная реализация систем шифрования Виженера и шифра «двойной квадрат» Уитстона. Было изучено задание, собрана информация, разработана структура, реализован и протестирован код, изготовлен готовый продукт.

Дополнительно ознакомился/изучил

Алгоритм шифрования Виженера и Уитстона, их реализация, история создания, достоинства и недостатки. Изучены новые способы построения программ  
и несколько библиотек для работы в Python

Заслуживает оценки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Руководитель практики  от предприятия  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_2023 г.  К.Ю. Гуфан |