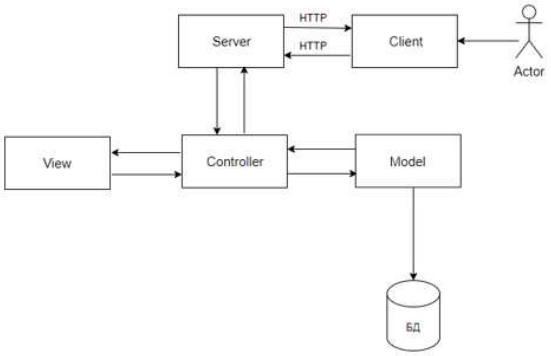

Рис. 2. Архитектура web-приложения

Клиент обращается к серверу HTTP запросом, затем сервер получает запрос и передаёт его на следующий уровень, уровень Controller. Controller отвечает за правильную обработку входящего запроса, а также определяет в каком представлении должны быть возвращены данные пользователю.

View преобразует полученные данные в нужный формат для отправки клиенту. Model обращается к базе данных за извлечением нужных данных. Затем пользователь видит на экране запрашиваемую информацию в корректном виде.

В завершении статьи хочется отметить, что разработка информационной системы, для автоматизации процесса нормоконтроля является необходимой вузу, так как позволит сократить время проверки учебной документации.

Список используемых источников:

1. Образовательный стандарт вуза ОС ТУСУР 01-2021. Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления от 25.11.2021. – URL: https://regulations.tusur.ru/documents/70
2. Введение в MVC – Электронный ресурс: сайт о программировании. – URL: https://metanit.com/sharp/ aspnet5/3.1.php

ШИФРОВАНИЕ И ДЕШИФРОВАНИЕ ТЕКСТА НА PYTHON

И.С. Аношин[а], студент гр. 17В11,
Научный руководитель: Разумников С.В [б], к.т.н., доц.
Юргинский технологический институт (филиал)
Национального исследовательского Томского политехнического университета,
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
E-mail: [а]Legenda.drovasek@gmail.com E-mail: [b]demolove7@inbox.ru

Аннотация: программа для шифрования и дешифровании текста на языке Python. Актуальность этой темы в том, что на сегодняшний день многие люди озабочиваются безопасностью и безопасной передачи данных. Для этого как раз подходит такой алгоритм шифрования, как RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) – алгоритм с открытом ключом, основывающийся на вычислении сложной задачи факторизации больших простых чисел.

Ключевые слова: Python, random, алгоритм, программа, блок-схема, разработка.

Abstract: a program for encrypting and decrypting text in Python. The relevance of this topic is that today many people are concerned about security and secure data transmission. For this, an encryption algorithm such as RSA (an abbreviation of the surnames Rivest, Shamir and Adleman) is just right – an algorithm with a public key based on the calculation of a complex problem of factorization of large primes.

Keywords: Python, random, algorithm, program, flowchart, developmentПрограмма должна показывать консольный интерфейс с различными вариациями режима работы. Например, главное меню программы, надёжность кода, показ инструкции и выход.

Генерация открытого ключа. Возьмём 2 разных простых числа (это такие числа, которые делиться только на себя и на единицу), допустим 233 и 521, назовём их q и p, далее перемножим их 233*521= 121 393 – это наш модуль, первая часть открытого ключа готова. Следующий шаг – нахождение фи, оно потребуется для нахождения экспоненты и секретного ключа. Но об этом позже, фи: (q-1)*(p-1) = (233-1)*(521-1) = 120 640. И наконец находим последнюю часть открытого ключа – экспоненту. Она должна быть простым числом, меньше фи и должна быть взаимно простой с фи. Возьмём, к примеру 211, 120 640 / 211 = 571,75355… то есть наше фи нельзя поделить на экспоненту без остатка. Итого наш открытый ключ – модуль: 121 393 и экспонента: 211.

Генерация приватного ключа. Возьмём модуль, фи, экспоненту и найдём вторую часть приватного (закрытого/личного) ключа. Наше d должно быть таким, чтобы перемножение его с экспонентой и делением с остатком на фи, равнялось единице. Только тогда мы сможем дешифровать наше сообщение, d = 39 451. Наш приватный ключ – модуль: 121 393 и d: 39 451

Теперь можем приступить к шифрованию и дешифрованию текста. Допустим, наше сообщение будет число 911 нам нужно его безопасно передать другому человеку так, чтобы это число не могли прочитать третьи лица. Для этого этот человек должен отправить нам открытый ключ (модуль: 121 393 и экспонента: 211). Далее мы шифруем наше сообщение: возводим его в степень экспоненты и результат делим с остатком на модуль: (911^211) %121 393 = 68 275 – это наше зашифрованное число, его то мы и передаём другому человеку. Он в свою очередь дешифрует его. Возводим его в степень d и результат делим с остатком на модуль: (68 275^39 451) % 121 393 = 911. Ну вот, всё получилось! Третьи лица, зная модуль и экспоненту не смогут (в кратчайшие сроки) расшифровать зашифрованное сообщение, ведь им надо знать q и p, а их можно получить только перебором. Ну да, 233 и 521 кажутся не такими большими, а если будет 998 443 и 998 471? Их модуль тогда будет 996 916 380 653!!! В общем, чем больше q и p, тем сложнее взломать ключ, тем больше времени потребуется третьим лицам для дешифровки данных. [3]

Описание алгоритма работы программы:

1. Человек выбирает пункты работы программы.
2. Если нужен открытый и приватный ключ.
   - 2.1 - Рандомно выбор двух чисел из списка простых чисел.
   - 2.2 - Расчёт модуля и числа фи.
   - 2.3 - Нахождение экспоненты (несколько раз в рандомном диапазоне).
   - 2.4 - Расчёт числа d.
   - 2.5 - Вывод в консоль открытый ключ (модуль и экспонента) и приватный ключ (модуль и d).
3. Если нужно зашифровать сообщение.
   - 3.1 - Ввод в консоль открытый ключ в консоль.
   - 3.2 - Ввод сообщения в консоль.
   - 3.3 - Перекодировка и шифрование текста.
   - 3.4 - Вывести результат в консоль.
4. Если нужно дешифровать сообщение.
   - 4.1 - Ввод приватного ключа в консоль.
   - 4.2 - Ввод сообщения в консоль.
   - 4.3 - Дешифрование текста с помощью d и модуля.
   - 4.4 - Вывод результата в консоль.
5. Если нужно зашифровать текст для себя.
   - 5.1 - То повторить предыдущие пункты.
6. Если нужна инструкция по работе программы.
   - 6.1 - Вывод в консоль инструкции.
7. Если нужно выйти из программы
   - 7.1 - Завершить работу

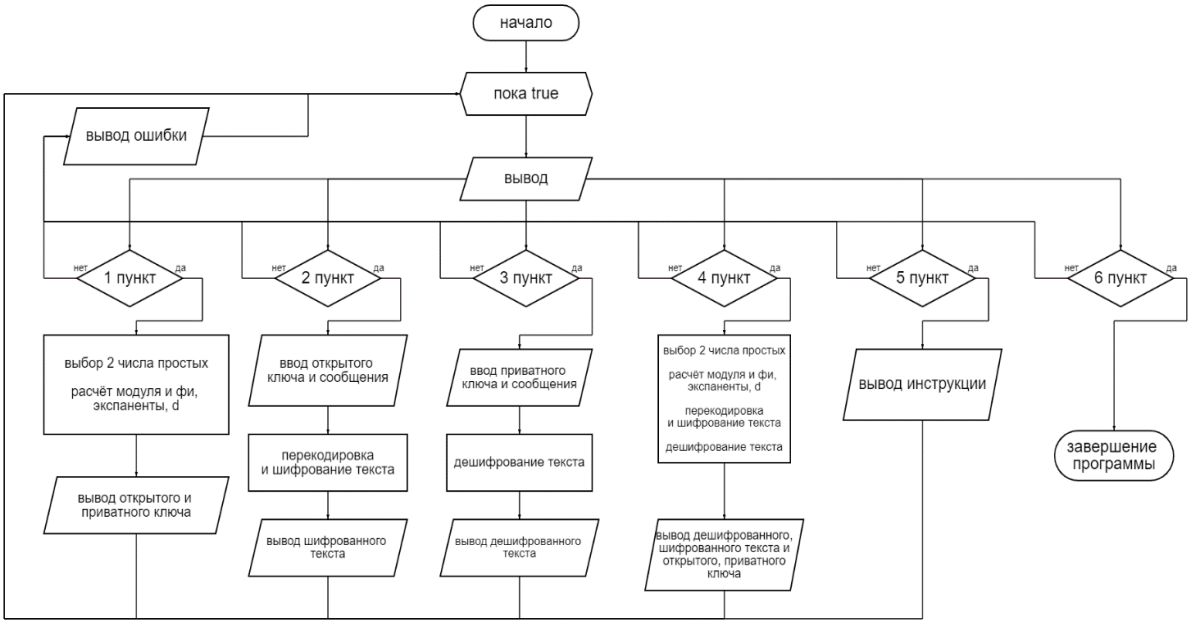

Рис. 1. Блок-схема всего алгоритма

Входными данными является print. Выходные данные представлены в виде текстовой информации, меню программы, инструкции, ключей, прогресс выполнения работы, шифрованный, дешифрованный текст.

В результате работы мы разработали надёжную, рабочую программу, которая реализует шифрование и дешифрование текста.

Список использованных источников:

1. Разумников С.В. Теория алгоритмов: методические указания к выполнению курсовой работы для студентов очной формы обучения, обучающихся по направлению 09.03.03 «Прикладная информатика» / С.В. Разумников – Юрга: Изд-во Юргинского технологического института (филиала) Томского политехнического университета, 2022. – 20 с.
2. Буйначев, С.К. Основы программирования на языке Python: учебное пособие / С.К. Буйначев, Н.Ю. Боклаг. – Екатеринбург: Изд-во Урал, ун-та, 2014. – 91, [1] с.
3. Алгоритм шифрования RSA: сайт e-nigma: – Электронный ресурс. – URL : https://e-nigma.ru/stat/rsa/ (дата обращения 18.01.2023)

АНАЛИЗ ВОЗМОЖНОСТЕЙ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБУЧЕНИЯ С УЧЕТОМ ОСОБЕННОСТЕЙ МЫСЛИТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ ОБУЧАЮЩИХСЯ

И.С. Аношин[a], студент гр. 17В11, А.В. Трофимов, студент гр. З-17Б91, А.П. Жолбин, студент гр. 17В11
Научный руководитель: Лизунков В.Г., к.пед.н. доц.,
Юргинский технологический институт (филиал)
Национального исследовательского Томского политехнического университета,
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
E-mail: [a]Legenda.drovasek@gmail.com

Аннотация: Анализ возможности повышение эффективности обучения с учетом особенностей мыслительной деятельности обучающихся. Актуальность данной работы обусловлена недостаточной изученностью повышения эффективности обучения с учетом особенностей мыслительной деятельности обучающихся.