где c_{kp}– стоимость маршрута *p* транспортного средства типа *k.*

При ограничениях на:

- запасы топлива на нефтебазах:

$$\sum_{(k,p)\in P} a_{kprj} x_{kp} \le D_{rj}\ ,\forall r,j\ , \tag{5}$$

гдеa_{kprj}– объем вывезенного топлива вида *r* с нефтебазы *j;*

- удовлетворение спроса АЗС:

$$d_{ri} \le \sum_{(k,p)\in P} b_{kpri} x_{kp} \le d_{ri} + \delta_{ri}\ ,\forall r,i\ , \tag{6}$$

где b_{kpri}– объем привезенного топлива вида *r*на АЗС *i,* δ_{ri} - заданные максимальные величины превышения спроса;

- назначение не более h_k ТС типа *k* на все маршруты:

$$\sum_{p\in M(P)} x_{kp} \le h_k\ ,\forall k\ \in K(P), x_{kp} \in Z, \forall (k,p)\in P\ , \tag{7}$$

гдеh_k – количество (идентичных) ТС типа *k*.

Таким образом, сформулирована задача целочисленного линейного программирования оптимизации маршрутов перевозки нефтетоплива, которая будет решаться в магистерской диссертации на тему «Метод поиска оптимальных маршрутов перевозки нефтетоплива».

ЛИТЕРАТУРА

1. А. Л. Неволина. Разработка метода планирования для нефтепродуктообеспечения сети автозаправочных станций. / А. Л. Неволина; ФГАОУ ВО “УФУ” Новосибирск 2017. – 34–35, 60 с.

2. В. В. Михайлович Алгоритмическая реализация численных методов / В. В. Михайлович; БНТУ – Минск, 2018: 32-38, 42-53 с.

3. О. В. Мартинсон. Моделирование автоматизации проектирования логических систем и потоковых процессов в строительстве / О. В. Мартинсон; ФГБОУ ВПО “МГСУ” – Москва, 2011: 1-2 с.

УДК 004.4

Студ. В.А. Баранов
Науч. рук. доц. Н.Н. Пустовалова
(кафедра информационных систем и технологий)

ШИФРОВАНИЕ ИНФОРМАЦИИ МЕТОДОМ СОЗДАНИЯ ДИНАМИЧЕСКОГО КЛЮЧА

Проблема защиты информации от несанкционированного доступа заметно обострилась в настоящее время в связи с широким распространением компьютерных сетей, особенно глобальных. Защита информации необходима для уменьшения вероятности разглашения,

утечки, умышленного искажения, утраты или уничтожения информации, представляющей определенную ценность для ее владельца.

Шифрование – это такое преобразование информации, которое делает исходные данные нечитаемыми и трудно раскрываемыми без знания ключа. Под ключом понимается секретная информация, определяющая, какое преобразование из множества возможных, выполняется в каждом конкретном случае над открытым текстом.

Основной принцип шифрования, предлагаемого в данной статье, заключается в том, что каждый символ получает свой уникальный код, который генерируется динамически по ходу процесса шифрования, т. е. обрабатывается отдельно, независимо от остальных.

Программа, реализующая шифрование информации методом создания динамического ключа, написана на языке программирования C++. Символ строки передается в функцию пользователя вместе с кодом шифрования, который получен с помощью встроенной функции random() и функции clock() стандартной библиотеки ctime [1]. В зависимости от алфавита (кириллица или латиница) и от регистра (верхнего и нижнего) символ будет шифроваться по-своему.

Рассмотрим известный шифр с названием ROT1. Этот шифр основан на принципе смещения символов (англ. rotate). Цифра после записи значит, насколько будет смещен символ по алфавиту вправо. Существуют и другие версии этого шифра (ROT2, ROT13 и т. д.).

В разработанной программе шифрование происходит по принципу шифра Цезаря, который является разновидностью шифра ROTn: исходный символ заменяется символом, стоящим в алфавитном порядке справа от исходного на количество, указанное заранее [2]. Поскольку символов в тексте может быть больше, чем в алфавите, то данная процедура замены продолжается нужное количество раз, используя символы алфавита заново.

Полученный результат записывается в динамический массив, который имеет размер такой же, как и исходная строка (на каждый символ по шифру). Ключ шифрования получают все символы в строке, однако шифруются только алфавитные символы. Формула шифрования имеет вид: (clock() + rand() % 10000) % (rand() % 1000).

Дешифрование – обратный шифрованию процесс. На основе ключа зашифрованный текст преобразуется в исходный открытый.

В разработанной программе может осуществляться и дешифровка с использованием сгенерированного шифра, однако этот шифр пропадает при закрытии программы, или при шифровании другого сообщения.

Использованный метод дешифровки делает невозможным чтение сообщения на другом девайсе. Чтобы дешифровать сообщение, следует ввести зашифрованную строку и ее шифр.

Таким образом, разработанный алгоритм, реализованный в программе, обеспечивает высокий уровень защиты данных против дешифрования и возможной модификации, защищенность информации основывается только на знании ключа, малое изменение исходного текста или ключа приводит к значительному изменению шифрованного текста, экономичность реализации алгоритма при достаточном быстродействии.

ЛИТЕРАТУРА

1. Н. В. Пацей. Основы алгоритмизации и программирования: учеб.-метод. пособие для студентов специальности «Информационные системы и технологии» / Н. В. Пацей. – Минск: БГТУ, 2010. – 289 с.

2. Ф. Циммерман. Введение в криптографию / Ф. Циммерман. – PGP Corporation, 2004. – 18 с.

УДК 004.4

Магистрант А.Г. Оганесян
Науч. рук. доц. Н.Н. Пустовалова
(кафедра информационных систем и технологий, БГТУ)

АНАЛИЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ PANDAS НА ЯЗЫКЕ PYTHON

Учет личных расходов является одним из инструментов оптимизации личных финансов. Система, которая смогла бы вести учет личных расходов и на основании собранных данных выдавать пользователю информацию о его тратах является актуальной и будет всегда востребована. Для обработки и анализа данных при решении данной задачи можно воспользоваться библиотекой pandas.

Pandas – это высокоуровневая Python библиотека для анализа данных. Она построена поверх низкоуровневой библиотеки NumPy, которая написана на языке Си, что является большим плюсом для производительности. С помощью pandas можно как анализировать данные, так и в связке с matplotlib визуализировать их.

Чтобы эффективно работать с pandas, необходимо освоить самые главные структуры данных библиотеки: DataFrame и Series[1].

Объект Series представляет собой объект, похожий на одномерный массив, но отличительной его чертой является наличие ассоциированных меток (индексов), вдоль каждого элемента из списка. Если индекс явно не задан, то pandas автоматически создаёт RangeIndex от