Вывод. Результаты проведенного исследования позволяют сформулировать следующие выводы:

- при представлении двигателя в виде инерционного звена первого порядка (без учета электромагнитного постоянного времени) результаты моделирования получаются не адекватными реальному процессу;
- оптимизация времени разгона двигателя 2ПН132М путем переключения напряжения питания привело к уменьшению длительности переходного процесса с 0,31с. до 0,104с.

Библиографический список:

1. Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. Изд. 4-е, перераб. и доп.- Спб.: Профессия, 2003 – 752с.
2. Электронный ресурс: http://leg.co.ua/knigi/oborudovanie/neispravnosti-elektrooborudovaniya-i-sposoby-ih-ustraneniya-15.html
3. В. И. Гуков, С.Ф. Позднухов, Н. Н. Рудано, В. Н. Тарасов. Авторское свидетельство № 997214 от 18.02.83г.**:** Способ пуска гистерезисного электродвигателя.

References:

1. Besekersky V. A., Popov E. P. Theory of automatic control systems. Ed. 4th, Rev. and extra - SPb.: Profession, 2003 – 752 p.
2. Electronic resource: http:. Leg.co.ua.knigi.oborudovanie/neispravnosti-elektrooborudovaniya-i-sposoby-ih-ustraneniya-15.html.
3. V. I. Gukov, S. F. Pozdnukhov, N. N. Rodano, V. N. Tarasov. Copyright certificate № 997214 from 18.02.83 G.: the Method of starting hysteresis motor.

УДК 519.14

Кадиев П.А., Кадиев И. П., Мирзабеков Т. М.

ПАКЕТ ПРОГРАММ ДЛЯ СКРЕМБЛИРОВАНИЯ ИНФОРМАЦИОННОГО ПОТОКА

Kadiev P. A., P. I. Kadiev, Mirzabekov T. M.

THE SOFTWARE PACKAGE FOR DATA STREAM SCRAMBLING

Аннотация. *Предлагается пакет программ для многовариантного ступенчатого преобразования текстового потока с целью повышения стойкости защиты от несанкционированного доступа, и пакет для восстановления преобразованного текста. В основе предложений: формирование пхп-массива из элементов потока данных, предварительные перестановки элементов массива, с формированием массива, каждая строка и каждый столбец которого включает один и только один элемент из*

каждой строки и каждого столбца исходного массива, с последующим считыванием по вариантам, выбранным пользователем.

Пакет для прямого преобразования включает в себя: модуль для формирования массива из входного потока; модуль перестановки элементов массива по схеме латинских квадратов; модуль считывания строк или столбцов массива по одному из следующих алгоритмов: последовательное считывание; считывание строк или столбцов с четными индексами, а затем с нечетными; считывание строк или столбцов с нечетными индексами, а затем с четными; считывание по случайному маршруту, который генерируется программой; считывание по маршруту, определенному пользователем.

Пакет для восстановления исходного сообщения путем обратного преобразования включает: модуль формирования канального массива из потока данных; модуль восстановления из канального массива – массива типа латинский квадрат; модуль восстановления исходного массива; модуль восстановления исходного сообщения.

Ключевые слова: *скремблирование, поток данных, информационный поток, ключи шифрования, дескремблирование.*

Abstract. *It is proposed a software package for multivariate stepwise transformation of the text flow in order to increase resistance to protect against unauthorized access, and a package to restore the converted text. The basis of the proposals: the formation of nxn-array from the elements of a data flow, preliminary transposition of the array elements to form an array, each row and each column of which includes one and one only element from each row and each column of the source array, following reading on the options selected by the user.*

Package for direct conversion includes: a module for forming an array from the input flow; transposition module of array elements according to the scheme of Latin squares; reading module of rows or columns of the array to one of the following algorithms: sequential reading; reading of rows or columns with even indices and then odd ones;reading the row or column with odd indices, and then the even; reading at random route, which is generated by the program; reading at the route determined by the user.

Package for restoring of the original message by the inverse transform comprises: a channel array forming module from the data flow; recovery module from the channel array - the array of Latin square type; the original array module; the original message restoring module.

Key words: *scrambling, data flow, information flow, encryption keys, descrambling.*

Введение. Скремблированием – процесс преобразование потока данных с целью видоизменения его структуры, заключающийся в перемежении символов, перестановке их местами. Это приводит к видоизменению структуры потока, внесению в нее свойств случайности, что может быть использовано: для защиты от несанкционированного доступа (НСД) к данным, скрытия его инфор-

мационного содержания, защиты от пакетов случайных ошибок или преднамеренных стираний при передаче по каналу, для улучшения спектральных и статистических характеристик потока. [1].

Многообразие решаемых скремблированием задач, обеспечивающих качество и эффективность информационных процессов, определяет множество методов ее реализации и необходимость разработки новых технологий, в частности, технологий, которые имели бы универсальный характер.

Постановка задачи. Выбор метода скремблирования во многом определяется спецификой решаемых задач. Поэтому часто методы скремблирования являются узко специализированными, что можно отметить в качестве их недостатка.

Наиболее распространенными из них являются: блочные периодические, сверточные и псевдослучайные. В простейшем варианте периодические реализуют детерминированное скремблирование путем записи кодовых блоков в столбцы (строки) матрицы со считывание по строкам (столбцам).

При псевдослучайном скремблировании в блоках элементы цифрового потока переставляются псевдослучайным образом, адрес переставляемого элемента генерируется генератором псевдослучайных последовательностей. При использовании сверточного метода скремблирования элементы потока логически складываются по модулю два с элементами пседослучайного потока. При использовании скремблирования для защиты от НСД метод периодических перестановок не применяется ввиду того он недостаточно криптостойкий и легко раскрываемый. При защите от пакетов ошибок он предпочтителен ввиду простоты реализации.

Методы свертки и псевдослучайных перестановок сложны, так как при их реализации нужных генераторы псевдослучайных последовательностей, требующие обеспечения их синхронности и синфазности. Однако их использование неизбежно, так как позволяет более эффективно решать задачи защиты от НСД и защиты от пакетов ошибок. Особенно это важно при борьбе с пакетами случайных ошибок, характеристики которых в каналах могут меняться, и при защите от преднамеренно организованных для стирания информации при передаче по каналу помех.

Из изложенного следует, что целесообразно разработать технологии с гибкой структурой, позволяющей формировать процессы скремблирования в зависимости от решаемых задач.

В данной работе качестве основного требования к процессу скремблирования рассматривается обеспечение надежной защиты от НСД. Требования по защите от пакетов случайных или организованных ошибок, рассматриваются с точки зрения изменения их свойств и характеристик.

Задача, поставленная в данной работе состоит в разработке пакета программ прямого и обратного преобразования потока данных, который обеспечивает улучшение характеристик потока:

– путем формирования массива из блоков, образованных из элементов потока данных;

- преобразованием массива, перед считыванием данных, перестановками элементов, используя метод перестановок в качестве первого элемента ключа шифрования;
- в полученном после преобразования массиве, обеспечить считывание по различным алгоритмам и маршрутами, которые выбирает пользователь, используя выбор метода в качестве дополнительного, второго ключа шифрования.

Пакет программ должен обеспечить:

- формирование из блоков потока данных заданной длины **nxn** – массив;
- перестановки элементов **nxn** – массива, по заданным алгоритмам;
- последовательное считывание содержимое полученного массива последовательно по строкам или, столбцам, в заданной пользователем последовательности, по псевдослучайному алгоритму
- обратные преобразования по восстановлению структуры исходного потока.

Методы исследования. В основе методов решения поставленной задачи - создание пакета программ скремблирования, реализующих процесс на основе теоретических результатов, полученных авторами в работах [2, 3] в качестве методов преобразования, сформированных из элементов потока данных массива и первого ключа шифрования, с последующим использование в качестве второго ключа шифрования одного из вариантов маршрута считывания содержимого преобразованного массива.

В качестве правил перестановок сформированного **nxn** – массива использован алгоритм, формирующий перестановками элементов массив, в каждой строке и в каждом столбце которого находится один и только один элемент из каждой строки и каждого столбца исходного массива [2, 3].

Искажения в таком массиве, от случайных пакетов ошибок или организационных помех стирания, длиною в n символов вызывает при обратном преобразовании на входе одиночные ошибки в строках, что легко обнаруживается и устраняется относительно простыми кодами Хэмминга.

Полученный перестановками массив, сохраняет указанные выше свойства при любых перестановках строк и столбцов. Это позволяет перед считыванием содержимого перестановками строк и столбцов сформировать **n!(n-1)!** различных по местоположению элементов массивов. Что означает наличие указанного количества ключей шифрования, и определяет криптостойкость первого уровня метода шифрования. Считывание может выполнено по любому из указанных вариантов полученных массивов.

Выбор метода считывания задается пользователем. Он может быть реализован множеством вариантов: по строкам в той последовательности, в которой они записаны в массиве, либо в любом из **n!** варианте считывания строк, определяемых пользователем выбором соответствующего ключа шифрования, задающего этот порядок.

Аналогично считывание может быть организовано считыванием по столбцам, причем число вариантов равно числу возможных перестановок столбцов.

Псевдослучайное скремблирование реализуется на уровне организации псевдослучайного выбора порядка считывания строк или столбцов, в отличие от традиционного случайного выбора номера позиции считываемого элемента в блоках информационного потока.

Для реализации метода в пакете использована стандартная программа формирования псевдослучайных последовательностей, определяющая порядок считывания строк или столбцов в преобразованном массиве.

Обсуждение результатов. В данной статье представлен пакет программ для скремблирования информационного потока, отличающийся тем, что сформированный из блоков потока данных информационный массив преобразуется по оригинальному алгоритму перестановок элементов, предложенному в работах авторов [2, 3], формирующему конфигурацию, строки и столбцы которого содержат по одному элементу из каждого столбца и каждой строки исходного массива, с последующей перестановкой ее строк и столбцов, считыванием их по различным алгоритмам, определяемым пользователем. Это позволяет существенно повысить криптостойкость метода шифрования.

Сам процесс скремблирования является многоступенчатым, число ключей шифрования значительно увеличивается.

В пакет включены программы формирования из входных информационных потоков массивов в виде ***nxn-*** матриц, где ***n*** = (3, 5, 7, 9,...), преобразования в матрицы, в строках и столбцах которых расположены по одному элементу из каждой строки и каждого столбца исходного массива [2, 3].

Содержимое пакета программ. Для скремблирования входного сообщения разработаны следующие модули:

- модуль формирования ***nxn-***массива из входного потока;
- модуль перестановки элементов массива по алгоритму формирования массив, строки и столбцы которого содержат по одному элементу из каждой строки и каждого столбца исходного ***nxn-***массива;
- модули считывания строк или столбцов массива по одному из следующих алгоритмов:

а) последовательное считывание;

б) последовательное считывание строк или столбцов с четными индексами, а затем с нечетными;

в) последовательное считывание строк или столбцов с нечетными индексами, а затем с четными;

г) считывание строк или столбцов по случайному маршруту, который генерируется программой;

д) считывание строк или столбцов по маршруту, определенному пользователем.

Программный пакет для восстановления скремблированного сообщения включает в себя:

- модуль формирования из потока данных канального массива;
- модуль восстановления из канального массива - массива типа латинский квадрат;
- модуль восстановление исходного массива;
- модуль восстановление исходного сообщения.

Разработанные интерфейсы приложений для скремблирования и дескремблирования информационного потока, отображены на рисунках 1 и 2.

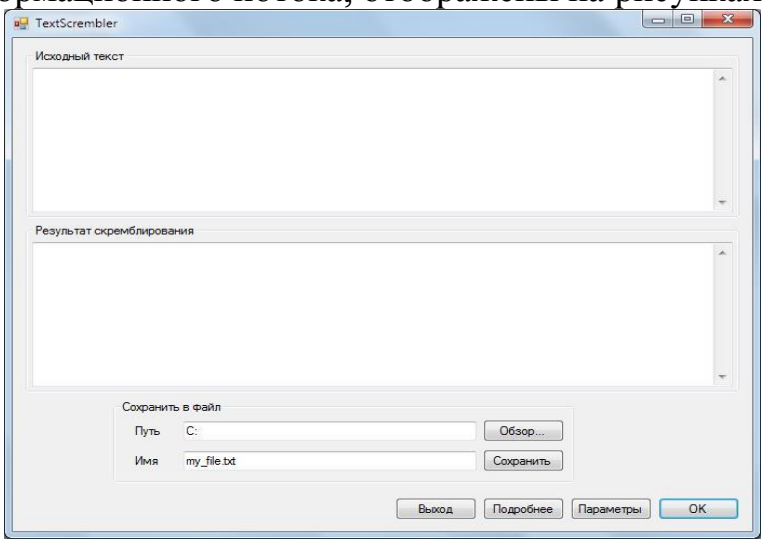

Рисунок 1 – Интерфейс программы для скремблирования информационного потока

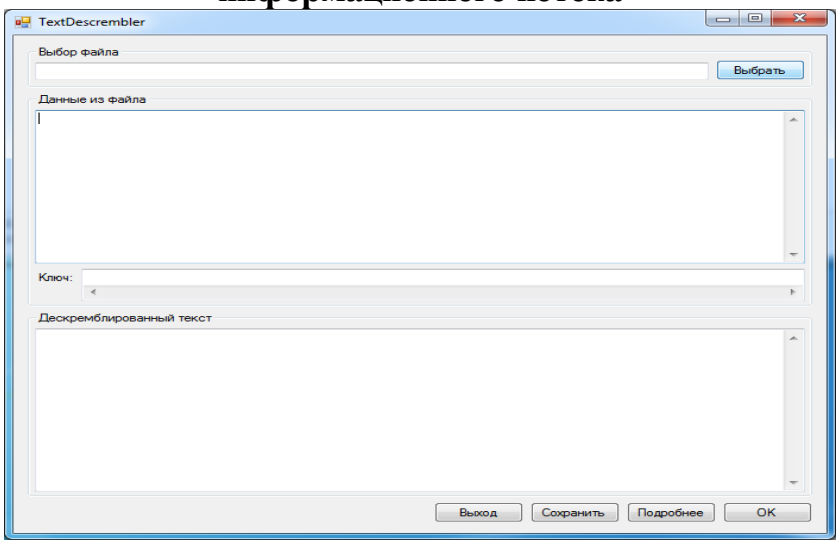

Рисунок 2 – Интерфейс программы для восстановления исходного сообщения

Процедуры по прямым преобразованиям выполняются в следующем порядке:

1. Сообщение, которое подлежит преобразованию, вводится в текстовое поле «Исходный текст». Для наглядности, в качестве исходного сообщения использована часть русского алфавита: «АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШ..» (рисунок 3).

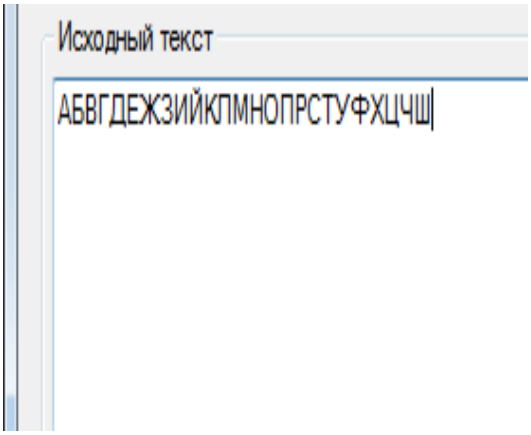

Рисунок 3 – Ввод исходного сообщения

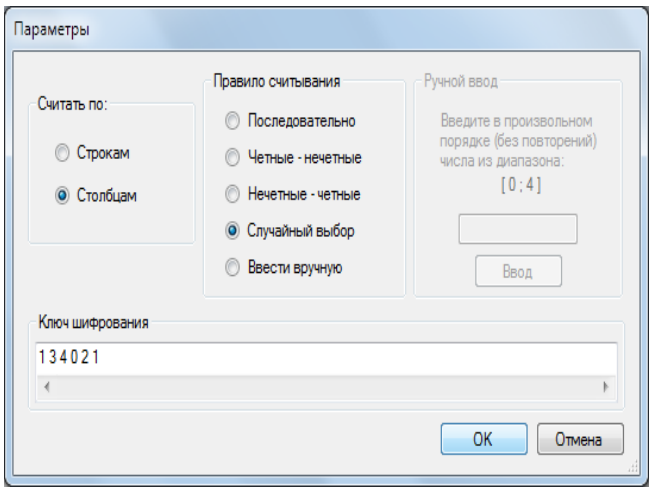

Рисунок 4 – Интерфейс окна «Параметры» программы для скремблирования информационного потока

2. Далее, при нажатии на кнопку «Параметры», всплывает диалоговое окно, в котором можно выбрать прядок и правило считывания (рисунок 4).

Интерфейс окна «Параметры» разделен на четыре блока:

Первый блок – «Считать по:». Здесь, пользователь задает порядок считывания: по строкам (по умолчанию), или по столбцам;

Второй блок – «Правило считывания», предназначен для выбора одного из следующих правил считывания:

а) «Последовательно» – является простейшим из всех возможных случаев, т. е. строки или столбцы будут переданы последовательно, без перестановок, от 0 до *N* следующим образом: 0, 1, 2, 3, ..., *N*-1, *N*;

б) «Четные-нечетные». В данном случае строки (столбцы) будут переставлены таким образом, что вначале будут строки (столбцы) с четными индексами, а затем с нечетными. Так как *N* – всегда нечетное, последовательность будет выглядеть следующим образом: 0, 2, 4, ..., *N*-3, *N*-1, 1, 3, ..., *N*-2, *N*;

в) «Нечетные-четные». Данное правило, является похожим на предыдущий случай, с единственной разницей в том, что в отличие от считывания по правилу «четные-нечетные», в начало будут переставлены строки (столбцы) с нечетными индексами, а затем – с четными. Так как *N* – всегда нечетное, последовательность будет выглядеть следующим образом: 1, 3, ..., *N*-2, *N*, 0, 2, 4, ..., *N*-3, *N*-1;

г) «Случайный выбор». Суть данного правила заключается в генерации случайной последовательности неповторяющихся чисел от 0 до *N*-1. Далее происходит перестановка строк или столбцов по сгенерированной последовательности;

«Ввести вручную». При выборе данного правила, пользователю необходимо будет ввести маршрут, по которому будут переставлены строки или столбцы;

Третий блок – «Ручной ввод». По умолчанию этот блок заблокирован. Для активации – необходимо выбрать правило считывания «Ввести вручную». Данный блок позволяет пользователю самостоятельно задавать последовательность считывания;

Четвертый блок – «Ключ шифрования». Отображает в текстовом поле ключ шифрования. Ключ состоит из N+1 чисел. Первое число ключа соответствует выбранному порядку считывания:

а) ноль (0) – если считывается по строкам;

б) единица (1) – если считывается по столбцам.

Остальные N чисел ключа, со второго и до N+1 элемента соответствуют правилу, по которому были переставлены строки или столбцы массива.

3. После выбора прядка и правила считывания, необходимо нажать на кнопку «ОК» в диалоговом окне программы, в результате запустится процесс скремблирования. Результат преобразования отобразится в текстовом поле «Шифрованный текст» как показано на рисунке 5.

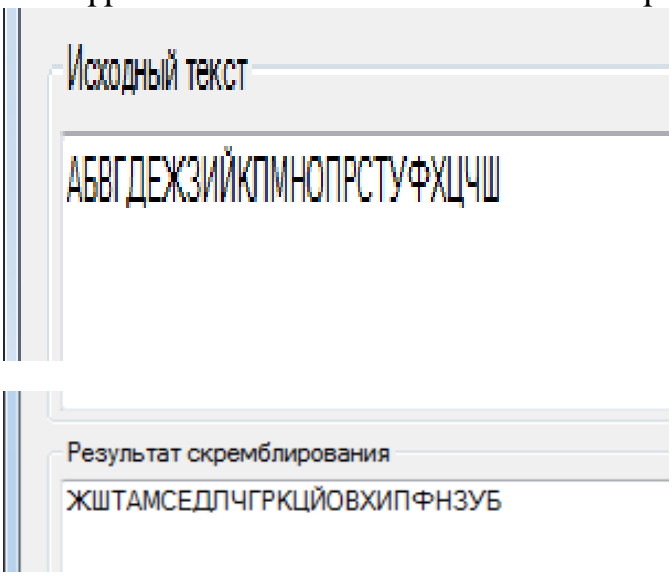

Рисунок 5 – Вывод результата скремблирования

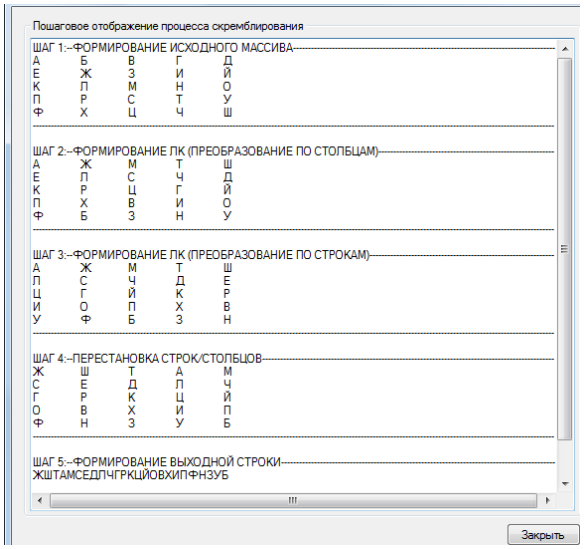

Рисунок 6 – Интерфейс окна «Подробнее» программы для скремблирования информационного потока

При нажатии на кнопку «Подробнее», появится диалоговое окно, в котором можно будет изучить каждый шаг преобразования исходного сообщения в скремблированный текст (рисунок 6).

4. После завершения скремблирования, результат необходимо сохранить в файл. Для этого в соответствующих полях интерфейса задается имя и путь к файлу, после чего нужно кликнуть по кнопке «Сохранить».

Восстановление скремблированного сообщения выполняется в следующем порядке:

1. Запускаем программу для восстановления (дескремблирования) исходного сообщения и загружаем файл со скремблированными данными, который был сохранен при работе предыдущей программы. Для этого нажмем

на кнопку «Выбрать» в основном окне программы для дескремблирования сообщения (рисунок 2) и в открывшемся окне найти и загрузить нужный файл.

2. После выбора и загрузки файла, программа считает данные из нее и отобразит их в соответствующих элементах интерфейса: в текстовое поле «Выбор файла» выводится полное имя выбранного файла; в поле «Данные из файла» – скремблированный текст; в поле «Ключ:» запишется информация, которая соответствует выбранному правилу преобразования-скремблирования (рисунок 7).

Окончательное восстановления исходного сообщения происходит при нажатии на кнопку «ОК» в основном окне программы для дескремблирования сообщения. Результат отобразится в поле «Дескремблированный текст» (рисунок 8).

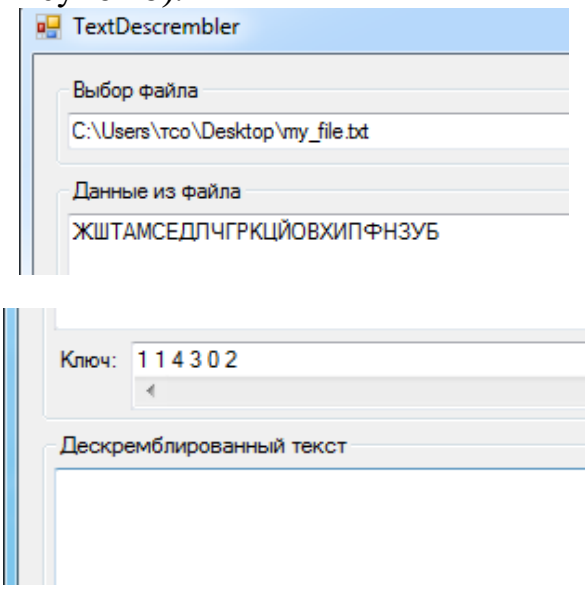

Рисунок 7 – Загрузка данных из файла в диалоговое окно программы

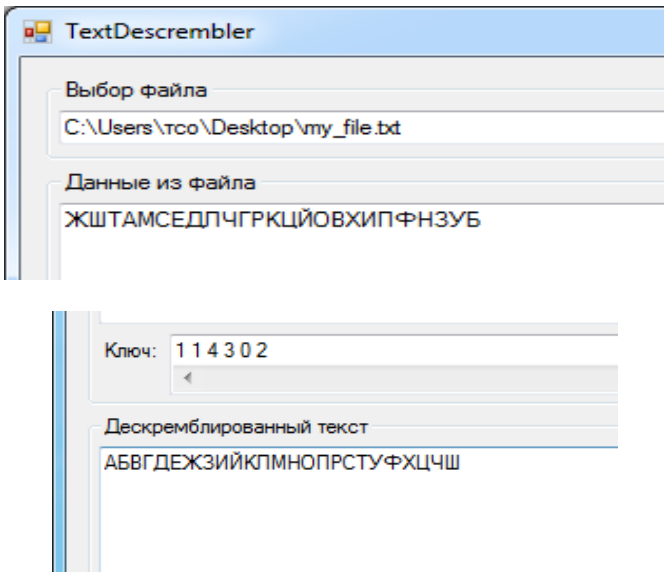

Рисунок 8 – Результат дескремблирования

3. Также как и в первой программе, здесь имеется возможность изучить каждый шаг преобразования-дескремблированния данных в исходное сообщение, используя кнопку «Подробнее».

На рисунке 9 видно, что восстановление исходного сообщение происходит по следующему алгоритму:

- восстановление массива из скремблированного текста (рисунок 8, шаг 2);
- обратная перестановка строк (столбцов) (рисунок 8, шаг 3);
- обратное преобразование по схеме латинских квадратов (рисунок 8, шаг 4 и 5);
- восстановление исходного сообщения (рисунок 8, шаг 6).

4. Для того чтобы сохранить в файл восстановленное сообщение необходимо нажать на кнопку «Сохранить» в основном окне программы для дескремблирования сообщения (рисунок 2), при этом, в том же каталоге, откуда был

загружен файл со скремблированным сообщением, появится новый файл, в который будет записано восстановленное исходное сообщение.

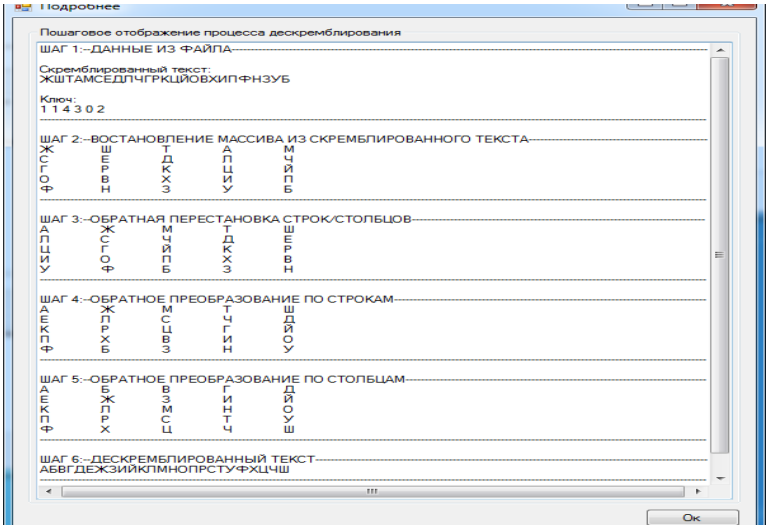

Рисунок 9 – Диалоговое окно «Подробнее»

В результате дескремблирования будет получено исходное сообщение: «АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШ».

Вывод. Представленный пакет программ разрабатывался в среде Microsoft Visual Studio 2010, используя язык программирования C#. Предлагаемый пакет является продолжением работ по созданию программ скремблирования на основе предварительного преобразования массивов по схеме двухиндексных латинских квадратов, начало которым было положено разработкой и регистрацией программы преобразования матриц методом латинских квадратов [4]. Пакет программ под общим названием «СКРЕМБЛИРОВАНИЕ» внедрен в учебный процесс в качестве лабораторной работы по дисциплине «Теория информации», подготовлен для Госрегистрации.

Библиографический список

1. Шевкопляс Б.В. Скремблирование передаваемых данных. [Электронный ресурс] – URL: http://lit.lib.ru/s/shewkopljas_b_w/text_0030.shtml. (дата обращения: 10.04.2016).

2. Кадиев П.А., Кадиев И.П., Зейналов М.З. Алгоритмы преобразования «классических» матриц в 2-х индексные латинские квадраты. Вестник ДГТУ, Технические науки, № 17, 2010. – С. 45-49.

3. Кадиев П.А., Кадиев И.П. Об одном классе комбинаторных конфигураций. Вестник ДГТУ, Технические науки, № 31, 2013. – С. 45-49.

4. Кадиев П.А., Зейналов М.З. Программа преобразования матриц методом латинских квадратов. Свидетельство о государственной регистрации программ для ЭВМ № 2009616143 от 09.11.2009.

References:

1. Shevkoplyas B.V. The scrambling of transmitted data. Available at: http://lit.lib.ru/s/shewkopljas_b_w/text_0030.shtml. (accessed: 10.04.2016).

2. Kadiev P.A., Kadiev I.P., Zejnalov M.Z. Algorithms of transformation of «classical» matrixes in two-index latin squares. Vestnik DSTU, Technical science, № 17, 2010. – pp. 45-49.

3. Kadiev I.P., Kadiev P.A. About one class of combinatory configurations. Vestnik DSTU, Technical science, № 31, 2013. – pp. 45-49.

4. Kadiev P.A., Zejnalov M.Z. Program for transformation matrices by Latin squares. State registration of computer programs certificate № 2009616143 from 09.11.2009.

УДК 697.9

Марченко А.С., Сулин А.Б.

ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЭЛЕМЕНТА СИСТЕМЫ ЖИЗНЕОБЕСПЕЧЕНИЯ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Marchenko A.S., Sulin A.B.

LOGIC SIMULATION OF LIFE SUPPORT SYSTEM COMPONENT IN REAL TIME

Аннотация. *В статье предложено использование методологии имитационного моделирования для оценки эффективности ступенчатого регулирования скорости вращения двигателя вентилятора при поддержании в заданных границах объемного расхода воздуха системы «вентилятор-фильтр». Приведен подробный алгоритм работы программы, составленной на основе прикладного пакета Any Logic. Анализируется возможность использования предложенного метода при проектировании систем вентиляции. Предложенная в статье методика позволяет на этапе проектирования определить максимальные интервалы замены фильтрующих элементов систем, а также спрогнозировать время необходимого переключения скоростей работы двигателя вентилятора. Использование методики позволяет отказаться от сложных систем поддержания постоянного расхода воздуха и максимально увеличить срок службы комплекта фильтрующих элементов.*

Методика логического моделирования процессов позволяет снизить затраты на строительство и повысить энергоэффективность зданий.

Ключевые слова: *системы вентиляции, системы очистки воздуха, имитационное моделирование, характеристика сети.*

Abstract. *The article proposed the use of simulation methods for evaluating the effectiveness of a stepped fan engine speed control while maintaining the air flow*