DOI: 10.17117/na.2014.01.191

Поступило в редакцию: 27.11.2014

<http://ucom.ru/doc/na.2014.01.191.pdf>**Димитриев А.П.****Схема работы криптосистемы на основе
задачи размещения и ее новая версия****Dimitriev A.P.****A diagram of the cryptosystem based on
the location problem and its new version**

В работе исследуются эксплуатационные характеристики комплекса программ по шифрованию данных и криптографическая стойкость применяемого алгоритма. На усмотрение пользователя можно в зашифрованных файлах дописывать заголовок или обходиться без заголовка. Для ускорения работы программы в новой версии используется длина блока 12 вместо 24, как это было в первой версии. Рассмотрена необходимость значительных изменений подстановочной последовательности. Изменен интерфейс пользователя программы

Ключевые слова: шифрование, криптографическая система, программа, алгоритм

Димитриев Александр Петрович*Кандидат технических наук, доцент**Чувашский государственный университет им. И.Н. Ульянова**г. Чебоксары, Московский пр-т, 15*

In this paper the performance characteristics of software package for data encryption and cryptographic strength of the applied algorithm are investigated. At the discretion of the user in an encrypted file is possible to add a header or do without header. To accelerate the work program of the new version has a block length of 12 instead of 24, as it was in the first version. Considered the need for significant changes wildcard sequence. Changed the user interface of the program.

Key words: encryption, cryptographic system, program, algorithm

Dimitriev Alexander Petrovich*Candidate of Technical Sciences, Associate Professor**Chuvash state university named I N. Ulyanov**Cheboksary, Moskovskiy ave., 15*

Как известно, информация является важнейшим понятием, с которым оперируют люди, причем у каждого могут быть свои интересы, которые могут послужить мотивами к ее хищению. Одним из аспектов защиты информации является конфиденциальность, то есть доступность только легальным пользователям [8, с. 469]. Один из методов обеспечения конфиденциальности информации – шифрование. Зашифрованная информация обычно хранится в компьютере в виде зашифрованных данных. Эти данные могут быть доступны для считывания стандартными методами, но без знания пароля, которым обладают только легальные пользователи, информации будет минимум (например, только объем файла, время создания и метод шифрования).

Цель работы – презентация комплекса программ по шифрованию данных. Задачи: исследование эксплуатационных характеристик комплекса программ и криптографической стойкости применяемого алгоритма.

Как рассмотрено в [1 – 6], разработана программа шифрования файлов, реализующая криптографическую систему на основе задачи размещения [7, с. 46-47].

Интерфейс пользователя в новой версии представлен на рис. 1. Здесь используются сокращения для того, чтобы стоящий за спиной человек сразу не понял, чем занимается пользователь программы. А это часто случается: например, если рядом стоит сетевой принтер, к нему подходят коллеги за распечатками.

Рис. 1

Особенность новой версии в том, что на усмотрение пользователя можно в зашифрованных файлах дописывать заголовок или обходиться без заголовка. Достоинство файлов без заголовка в том, что при использовании после шифрования дополнительного внешнего шифратора становится невозможным подбор пароля для внешнего шифратора. Это обусловлено тем, что получаемый файл представляет собой псевдослучайный набор байтов, и даже если будет использован правильный пароль, это не будет видно даже при использовании статистического анализа.

Если же заголовок оставлять, то самому пользователю будет легче понять, что это за файл и какой он версии.

Скорость работы программы демо-версии оставляет желать много лучшего. Файл размером 209 кбайт шифровался 6 минут и столько же времени ушло на дешифрацию. Ускорение возможно путем исследования эффективности алгоритма при изменении конструктивных параметров.

Это сделано в демо-версии 1.2 той же программы. Новое здесь в использовании длины блока 12 вместо 24. Соответственно, скорость обработки информации увеличилась примерно в 2 раза. Так, файл размером 15 кбайт шифровался всего 7 секунд. Это связано, прежде всего, с временной сложностью алгоритма размещения, которая пропорциональна n^2 , где n – число объектов (искусственно создаваемых и описываемых по одному каждым двумя байта-

ми блока), в данном случае оно сократилось с 12 до 6. Однако, данная временная сложность лишь часть общей временной сложности алгоритма шифрования, остальная часть существенных изменений не претерпела, поэтому время сократилось на в 4, а в 2 раза. Единственно, что изменилось – устранено лишнее повторное вычисление итоговой целевой функции, но это особого ускорения не дало.

В основной версии скорость работы доведена до приемлемого значения – приблизительно в 40 раз быстрее, чем для демо-версии.

Рассмотрим вопрос о криптографической стойкости. Используется пароль, при этом известно, что обычно пароли короткие. Программа предусматривает длину пароля до 24 символов из набора ASCII, но в действительности многие символы этого набора не используются просто потому, что их трудно вводить и тем более запоминать.

Пароли необходимо защищать от компрометации. Дело в том, что человек не в состоянии запомнить много паролей (свыше 10-15) и поэтому часто использует их повторно для разных информационных систем. Если злоумышленнику известны исходное и зашифрованное сообщение, он может попытаться определить пароль, чтобы далее его использовать для раскрытия других сообщений. Поэтому нельзя слишком сокращать количество объектов в алгоритме программы и длину блока. Например, если длина блока будет равна четырем, то число объектов – всего два, и уже не так сложно по размещению и алгоритму определить изменения в подстановочной последовательности, особенно когда блоков много, и ко всем применяется один и тот же пароль, хэш от которого собственно и нужен для других сообщений.

Для представления об алгоритме шифрования введем следующие обозначения.

g – пароль, h – хэш от пароля, s – исходное сообщение, s'' – зашифрованное сообщение, t, p – векторы входных данных для размещения, b – вектор выходных данных при размещении, P – подстановочная последовательность, C – скремблирование, I – применение хэширования первый раз (см. рис. 2).

Рис. 2

Вначале (I раз, см. блок I) g играет роль двух составляющих, описывающих объекты – t , p . Если h становится известен, по нему невозможно определить g , даже если размерность вектора равна 1, согласно свойствам хэш-преобразователя. Из b определить из t, p невозможно по этой же причине, поэтому b искать не нужно. Размерность вектора выбирается большей для значительных изменений P , а это необходимо.

Доказательство необходимости значительных изменений P .

Если только незначительно изменять P , некоторые элементы s могут совсем не изменяться в s' (например, изменения происходят где-то на другом краю P). Когда s и s'' известны, а $s=s'$, можно найти h элементарной операцией «исключающее или». Достаточно нескольких совпадений h при разных s_i , чтобы с уверенностью сказать, что h найдено.

Если было 2 совпадения, это могло быть случайностью при длинном сообщении: раз в 256 случаев такое возможно при равномерном распределении вероятностей встречаемости символов. Тогда, если произвести статистический анализ, можно проверить правильность нахождения h .

При больших изменениях P совпадения маловероятны, а для таких изменений надо иметь большую размерность b . С другой стороны, если мало объектов, их взаиморасположение при размещении (b) становится предсказуемым, соответственно облегчается задача определения изменения P , а если известны P и s , снова становится известным s' .

Заключение

Часть исходного кода программы написана на ассемблере с целью высокой скорости работы, что делает исходный код аппаратно-зависимым. Интерфейс пользователя исправлен для интуитивно понятной работы.

** По вопросам получения вышеописанной программы просьба обращаться к автору данной статьи: **dimitrie1@yandex.ru***

Список литературы:

1. Димитриев А.П. Алгоритм сдвига для криптосистемы // Наука и образование в жизни современного общества. Ч. 1. Тамбов: Консалтинговая компания Юком, 2015. С. 65-67.
URL: <http://ucom.ru/doc/conf/2014.12.30.01.pdf>
2. Димитриев А.П. Алгоритм инициализации для криптосистемы // там же. С. 64-65.
3. Димитриев А.П. Алгоритм шифрования на основе задачи размещения // Современные тенденции в образовании и науке. Ч. 2. Тамбов: Консалтинговая компания Юком, 2014. С. 41-43.
URL: <http://ucom.ru/doc/conf/2014.11.28.02.pdf>
4. Димитриев А.П. Алгоритм дешифрования на основе задачи размещения // там же. С. 40-41.
5. Димитриев А.П. Константы для алгоритма шифрования на основе задачи размещения // APRIORI. 2014. № 6. С. 18. URL: <http://apriori-journal.ru/seria2/6-2014/Dimitriev1.pdf>

6. Димитриев А.П. Криптосистема с дайджестами, получаемыми в задаче размещения // Отечественная наука в эпоху изменений: постулаты прошлого и теории нового времени.

Ч. 4. № 4. 2014. С. 8-11.

URL: http://national-science.ru/files/Arhiv/7-8.11.2014/National_4_p4.pdf#page=8

7. Димитриев А.П. Модели и алгоритмы в системах автоматизированного перевода текста // Прикладная информатика. 2013. № 6 (48). С. 45-59.

8. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2007. 539 с.

© 2014, Димитриев А.П.

Схема работы криптосистемы на основе задачи размещения и ее новая версия

© 2014, Dimitriev A.P.

A diagram of the cryptosystem based on the location problem and its new version