УДК 003.26.09 : 511.13

В. Б. Иванчишин[1], Ю. Ф. Мухопад[1]

[1]Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация

КОДИРОВАНИЕ ЗАЩИЩАЕМЫХ МАССИВОВ ИНФОРМАЦИИ НА ОСНОВЕ ЧИСЛОВЫХ СООТНОШЕНИЙ

Аннотация. *Предложен метод кодирования информации на основе десятичной системы счисления. Кодирование символов (знаков) открытых данных осуществляется посредством индивидуальных (т.е. присущих кодируемому символу) цифровых кодов из x цифр, где x ≥ 2. Для распределения цифровых кодов и их применения при кодировании избираются цифровые последовательности десятичных периодических дробей. Один и тот же символ, повторно встречающийся в кодируемых данных, получает новый цифровой код. Количество различных цифровых кодов для кодирования любого повторно встречающегося символа определяется количеством цифр в цифровом коде. При кодировании символа тремя цифрами количество различных кодов для любого символа составляет 1000; при кодировании четырьмя цифрами – 10 000. Основы алгоритма изложены на естественном языке, что позволяет разработать программное обеспечение для обмена информацией, применяя быстродействующие вычислительные комплексы.*

Ключевые слова: *криптография, периодические дроби, кодирование информации, защита информационных массивов, криптостойкость.*

V. B. Ivanchishin[1], Yu. F. Mukhopad[1]

[1] Irkutsk State Transport University, Irkutsk, Russia

CODING OF THE PROTECTED ARRAYS OF INFORMATION ON THE BASIS OF NUMERICAL RATIOS

Summary. *The method of coding of information on the basis of a decimal numeral system is offered. Coding of symbols (signs) of open data is carried out by means of individual (i.e. inherent in the coded symbol) digital codes from x figures where x ≥ 2. For distribution of digital codes and their application when coding the digital sequences of decimal periodic fractions are chosen. The same symbol which is repeatedly found in the coded data receives a new digital code. The quantity of various digital codes for coding of any repeatedly found symbol is defined by the number of figures in a digital code. When coding a symbol in three figures the quantity of various codes for any symbol is 1000; when coding in four figures – 10 000. Bases of an algorithm are stated in a natural language that allows to develop the software for exchange of information, using high-speed computer systems.*

Keywords: *cryptography, periodic fractions, coding of information, protection of information massifs, cryptofirmness.*

Введение

Защита информации в настоящее время приобретает исключительно важное значение в экономической, политической, социальной и военной сферах деятельности [1-5].

Многие способы защиты информации основаны на модификации известных алгоритмов или, являясь оригинальными, не публикуются в открытой печати. В настоящее время наметились два подхода к построению средств защиты информации: 1) создания аппаратных средств с уровнем криптостойкости, приближающимся к предельно возможному [6-9], предназначенных для быстродействующих систем передачи и приема информации в реальном времени; 2) создание алгоритмических систем высокой сложности, ориентированных на применение вычислительных сверхбыстродействующих комплексов. Исследования в этих направлениях по-прежнему актуальны.

Предлагаемый метод кодирования информации

В данной статье рассматривается новый подход к разработке алгоритмов защиты информационных массивов текстового сообщения. Реализация этого подхода требует

применения быстродействующих средств вычислительной техники со специальным программным обеспечением.

Способ определения зашифрованных цифровых комбинаций определяет последовательность нескольких действий. Рассмотрим последовательность таких действий на упрощенном примере применения цифрового кодирования русского текста:

1. В русском тексте 32 буквы и 8 служебных символов (. , : ; ? ! - …),- всего 40 знаков (символов) открытых данных. Выделим из 40 символов подмножество из 10 символов либо по определенному семантическому принципу, либо из первых 10 букв алфавита: а, б, в, г, д, е, ж, з, и, й. Семантический принцип это соответствие какому-то смыслу. Например, слово «благостный» состоит из 10 разных букв т.е. будет использован искусственный «благостный» язык.

2. Десять выше избранных букв можно кодировать разными числами, состоящими из двух или более цифр. При двух цифровом кодировании каждому символу будет соответствовать код из двух цифр. Т.к. всех цифр 10 (от 0 до 9), то легко установить количество всех цифровых перестановок при двух цифровом кодировании. – Это количество равно последовательности из 100 чисел: 00,01,02… 98 и 99 (т.е. простая числовая последовательность от 00 до 99 включает все множество перестановок из 10 цифр, ибо не пропущено ни одного числа из двух цифр!). При трех цифровом кодировании последовательность всех перестановок составит 1000 т.е. последовательность 1000 чисел от 000, 001, 002…до 998 и 999. Формула количества всех цифровых перестановок при кодировании каждого символа открытых данных x цифрами – 10^x, где x – количество цифр, кодирующих 1 символ. При кодировании символа двумя цифрами данная формула дает 10^2=100; при кодировании символа 10 цифрами - 10^{10} =10 000 000 000.

Упрощая изложение рассмотрим двух цифровое кодирование десяти выше избранных букв. Исходные данные для двух цифрового кодирования можно представить в виде 100 двух цифровых столбцов, представляющих последовательность 100 чисел: 00, 01, 02, 03 и т.д. до 99. Каждый столбец обозначим буквой А с соответствующим числовым индексом: $A_1 A_2$ A_2 и т.д. A_{100}:

0 0 0 0 0 0 0 и т.д. 9 9 9 } 2-х цифровая исходная числовая
0 1 2 3 4 5 6 и т.д. 7 8 9 последовательность цифровых кодов
$A_1A_2A_3A_4A_5A_6A_7$ и т.д. A_{98} A_{99} A_{100}} обозначения цифровых столбцов с последовательной числовой индексацией

Последовательность цифровых столбцов A_i составляет совокупность *элементов* цифрового кодирования. На их основе можно осуществить кодирование текста, изложенного на естественном языке. При этом любая буква, в принципе, может кодироваться любым из 100 цифровых столбцов.

Если 10 избранных букв последовательно записывать над 100 цифровыми столбцами, то эти 10 букв можно повторить 10 раз:

а б в г д е ж з и й а б в и т.д. з и й
$A_1A_2A_3A_4A_5A_6A_7A_8$ A_9 $A_{10}A_{11}A_{12}$ A_{13} и т.д. A_{98} A_{99} A_{100}

Получена исходная база данных для двух цифрового кодирования избранных символов (букв). В представленном построении любая из 10 букв может быть закодирована 10 разными цифровыми кодами. Так, девятую букву **и** 10-буквенного алфавита можно кодировать столбцами: A_9 (соответствует цифровому коду 08), либо A_{19} (соответствует цифровому коду 18), либо далее A_{29}, A_{39}…A_{99} (соответствует предпоследнему элементу, имеющему цифровой код 98). Если же буква в кодируемом тексте повторится более 10 раз, то ей должен быть *повторно* присвоен один из ранее использованных кодов.

Введем правило: с каждым повторным появлением символа (буквы) в кодируемом тексте присваивать символу новый цифровой код из новой последовательности цифровых кодов, соответствующих повторной последовательности кодируемых символов. – При этом другие последовательные символы текста получают цифровые коды из той же новой последовательности цифровых кодов до тех пор, пока какой-либо из них не встречается

повторно. Тогда его кодирование, как и последующих символов осуществляется по другой последовательности цифровых кодов, соответствующей очередному повтору алфавита кодируемых символов.

Рассмотрим кодирование на примере *условного* отрывка текста: «виги гиби», применяя данную выше исходную базу данных для двух цифрового кодирования 10 избранных букв. Первые 3 буквы встречаются однократно, поэтому получают коды из первых 10 столбцов цифровых кодов, соответствующих первому распределению избранного алфавита: «виг» - A_3 A_9 A_4 (или в двух цифровом кодировании – 020803). Далее буквы «и г» встречаются 2-ой раз, поэтому по принятому правилу кодируются из второго распределения алфавита над последовательностью вторых 10 столбцов цифровых кодов: «иг» - A_{19} A_{14} (1813). В последующем буквенном сочетании «иб» буква и встречается 3-ий раз, а б – 1-ый, поэтому следует перейти к кодированию по третьей последовательности алфавита над цифровыми кодами т.е. «иб» получает коды A_{29} A_{22} (2821). Наконец, заключительная буква и получает цифровой код из 4-ой последовательности 10 цифровых столбцов – A_{39} (38). Итак, отрывок текста после кодирования принимает вид: 0208031813282138. Авторизованный абонент, имея такую же, исходную базу данных для двух цифрового кодирования легко декодирует зашифрованный текст т.к. коды 020803 принадлежат первому интервалу распределения кодируемых символов (букв) и при декодировании переводятся «виг»; коды 1813 принадлежат второму интервалу распределения символов и, соответственно, переводятся «иг» и т.д.

Но данный пример весьма примитивен и будет мгновенно расшифрован с помощью современных средств и технологий дешифрования. Уже первичный анализ выявляет тенденцию увеличения числовых кодов (от 02 вначале до 38 в окончании), что предполагает применение последовательно возрастающих числовых кодов. Поэтому, зная с некой достоверностью естественный язык информационного обмена, *несанкционированный субъект* может легко дешифровать закодированную информацию, исходя из последовательного распределения цифровых кодов (от 00 до 99). – Отсюда вопрос: как повысить криптостойкость шифра [2, с.28]?

Подход к реализации данной мысли ставит вопрос: *каково количество видоизменений* исходной последовательности цифровых кодов путем различных перестановок её элементов?

Формула множества перестановок из N элементов по N [11, с .454]:

$$N! = N\cdot(N-1)\cdot(N-2)\cdot\ldots\cdot3\cdot2\cdot1$$

Для нашего примера N=100, поэтому множество всех возможных перестановок весьма велико: $100! = 100\cdot99\cdot98\cdot\ldots\cdot3\cdot2\cdot1$. Но большинство перестановок примитивны и потому «легко узнаваемы» при дешифровании. Так, можно поменять местами любые 2 элемента: A_1 и A_{100}, A_1 и A_{99} **…;** A_2 и A_{100}, A_2 и A_{99} …; можно поменять местами 4 элемента: A_1 A_2 и A_{100} A_{99} и т.д. Множество единичных или локальных перестановок практически не изменяют *первооснову* исходной последовательности цифровых кодов. – Применение таких примитивных перестановок для кодирования приводит к созданию шифра низкой криптостойкости. Для повышения криптостойкости шифра необходимо множественное и неочевидное «перемешивание» элементов исходной последовательности цифровых кодов.

Существенно повысить криптостойкость цифрового кодирования можно формируя новую последовательность цифровых кодов путем **неочевидного отбора цифровых столбцов** по цифровой последовательности избранной периодической дроби [11, с.455]. Поясним идею такого кодирования.

Соотношения между некоторыми натуральными числами выражают бесконечные периодические дроби. Для **«дроби a/p, где p-простое число и 1 ≤a ≤ p-1, длина периода является делителем p-1».** Число **a** может быть простым или составным. Например, период десятичной дроби соотношения 1/17 должен быть равен (17-1) = 16 цифр, а именно: 1/17=0,0588235294117647*05882*... . Отметим: период соотношения всегда четное число, но не всегда равен величине (p-1), а может быть кратно меньше расчетной величины. Так, период дроби 1/13 =0.0769230769*2*... имеет 6 цифр, а не 12. А период соотношения 1/173 равен

расчетному (173-1)=172 цифры. Т.е. необходима вычислительная проверка числового соотношения.

Различные числовые соотношения строго индивидуальны (так, 3/7 ≠ 5/7), поэтому индивидуальны периоды их десятичных дробей. Однако, в больших периодах вполне можно найти цифровые интервалы, повторяющие периоды малых цифровых последовательностей. Так же можно найти одинаковые цифровые цепочки для чисел 11487 и 361 531 148 732 и т.д.

На основе данной закономерности приведем пример формирования новой последовательности цифровых кодов путем неочевидного перемешивания цифровых кодов исходной последовательности:

1. упрощая изложение изберем простое соотношение 1/7= 0, 1428571428.., период которого равен (7-1)=6 цифр. Применим этот период для построения новой последовательности цифровых кодов;

2. построение производится циклически. *В первом цикле построения,* избрание цифровых столбцов производится из 100 столбцов *исходной последовательности*. Последовательность избрания столбцов определяет цифровая последовательность периода избранной по п. 1 дроби: 142857.

Избрание столбцов для формирования новой последовательности цифровых кодов определяет *последовательно возрастающая* сумма цифр периода избранной дроби: 1-ый столбец избирается по 1-ой цифре периода, 2-ой столбец избирается *по сумме двух первых цифр* периода, 3-ий – по сумме 3-х первых цифр и т.д. Цифры 0 из периода дроби «не участвуют» в избрании столбцов, ибо нули не изменяют сумму цифр.

Первый цикл построения завершается тогда, когда последовательно нарастающая сумма цифр максимально приблизится к числу 100 (количество столбцов для двух цифрового кодирования), а прибавление следующей цифры периода дает сумму больше 100. Если период избранной дроби невелик сравнительно с количеством элементов исходной матрицы (в примере сумма 6 цифр <<100), то сумма цифр периода даст величину существенно меньшую 100. При этом, для продолжения 1-го цикла построения новой последовательности цифровых кодов, к достигнутой сумме цифр последовательно прибавляются цифры 2-го периода избранной дроби и т.д.

3. Построим *новую последовательность* цифровых кодов:

3.1. первая цифра избранной дроби – 1, поэтому *первым цифровым столбцом* избирается 1-ый столбец *исходной последовательности* – $\mathbf{A_1}$ (цифровой код столбца 00). Сумма первой и второй цифр периода 1+4=5, поэтому вторым столбцом избирается 5-ый столбец – $\mathbf{A_5}$ (код 04). Третьим столбцом по сумме цифр 1+4+2=7 избирается 7-ой столбец исходной последовательности –$\mathbf{A_7}$ и, соответственно, далее столбцы $\mathbf{A_{15}, A_{20}, A_{27}}$. – При этом двадцать седьмым столбцом завершается отбор столбцов по 1-ому периоду дроби, поэтому для продолжения 1-го цикла отбора столбцов применяем цифровую последовательность 2-го периода дроби: 27+1=28 т.е. седьмым столбцом новой последовательности избирается 28-ой столбец исходной - $\mathbf{A_{28}}$ (код столбца 27) и далее $\mathbf{A_{32}}$, $\mathbf{A_{34}}$, $\mathbf{A_{42}}$, $\mathbf{A_{47}}$, $\mathbf{A_{54}}$, - при этом завершается построение по 2-му периоду избранной дроби и производится переход к 3-ему периоду с избранием следующих столбцов: $\mathbf{A_{55}, A_{59}, A_{61}, A_{69}, A_{74}, A_{81}}$. Завершение 1-го цикла построения новой *последовательности* цифровых кодов производится по 4-му периоду дроби: $\mathbf{A_{82}, A_{86}, A_{88}, A_{96}}$,- следующая цифра 4-го периода дроби - 5 делает сумму цифр больше 100 (96+5=101), поэтому 1-ый цикл построения завершается столбцом $\mathbf{A_{96}}$, соответствующим 4-ой цифре 4-го периода дроби. Отбор столбцов по 1-му циклу:

$\mathbf{A_1\ A_5\ A_7\ A_{15}\ A_{20}\ A_{27}A_{28}\ A_{32}A_{34}\ A_{42}A_{47}\ A_{54}\ A_{55}\ A_{59}\ A_{61}A_{69}A_{74}A_{81}A_{82}A_{86}\ A_{88}A_{96}}$

В итоге первого цикла из 100 исходных столбцов избрано 22 столбца (6·3+4), а другие 100-22=78 столбцов остались без применения, поэтому для их включения в *новую последовательность* необходим переход ко второму циклу построения, начинающемуся с 5-ой цифры 4-го периода.

3.2. Перед вторым циклом построения необходимо последовательно записать *оставшиеся без применения* 78 столбцов. Далее следует *дать* ***новую*** *последовательную нумерацию* этим 78 столбцам:

$A_2A_3A_4A_6A_8A_9A_{10}A_{11}A_{12}\ A_{13}\ A_{14}A_{16}\ A_{17}A_{18}A_{19}A_{21}A_{22}A_{23}\ A_{24}A_{25}A_{26}A_{29}A_{30}\ A_{31}A_{33}$
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

$A_{35}\ A_{36}\ A_{37}A_{38}\ A_{39}\ A_{40}\ A_{41}\ A_{43}\ A_{44}\ A_{45}\ A_{46}A_{48}A_{49}A_{50}\ A_{51}A_{52}A_{53}A_{56}A_{57}A_{58}\ A_{60}\ A_{62}A_{63}$
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

$A_{64}\ A_{65}A_{66}\ A_{67}A_{68}\ A_{70}A_{71}A_{72}A_{73}\ A_{75}\ A_{76}\ A_{77}A_{78}A_{79}\ A_{80}A_{83}A_{84}\ A_{85}A_{87}A_{89}A_{90}A_{91}A_{92}$
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71

$A_{93}\ A_{94}A_{95}A_{97}A_{98}A_{99}\ A_{100}$
72 73 74 75 76 77 78

Для перехода от одного цикла отбора столбцов к другому, необходимо прибавить к сумме цифр первого цикла следующую цифру соответствующего периода дроби. Новая сумма цифр станет больше количества столбцов, участвовавших в предшествующем отборе столбцов. Т.е. часть единиц из очередной цифры дроби дополнит сумму цифр до количества столбцов, участвовавших в предшествующем отборе столбцов, а по остающейся части единиц (≥1) будет отобран по новой нумерации порядковый столбец для следующего цикла построения цифровых кодов.

В данном примере 1-ый столбец 2-го цикла построения определяется прибавлением к сумме цифр 1-го цикла (96) 5-ой цифры 4-го периода дроби (14285**7**): 96+5=101. При этом, 4 единицы из 5 дополняют до 100 количество столбцов участвовавших в 1-ом цикле (96+4=100), а оставшаяся 1 применена ко 2-му циклу отбора из вновь пронумерованных 78 столбцов. Поэтому *первым столбцом* во 2-ом цикле отбора становится столбец $\mathbf{A_2}$. Далее последовательно прибавляя шестую цифру (7), затем цифры 5-го периода (142857) и др. периодов получим последовательность столбцов 2-го цикла построения новой последовательности цифровых кодов: 1+7=8 - 8-ой столбец по новой нумерации (A_{11}); далее 8+1=9 (A_{12}); и т.д. Второй цикл завершается суммой цифр – 77 (<78). В итоге отобраны следующие 18 столбцов из 78:

$A_2A_{11}A_{12}A_{17}A_{19}A_{30}A_{37}A_{45}A_{46}A_{51}A_{53}A_{65}A_{71}A_{79}A_{80}A_{87}A_{90}A_{99}$

Второй цикл завершается прибавлением 4-ой цифры **(8)**, 7-го периода дроби (сумма цифр - 77<78). Оставшиеся (78-18)=60 столбцов получают *новую* последовательную нумерацию и этим подготовлены к 3-ему циклу:

$A_3A_4A_6A_8A_9A_{10}A_{13}A_{14}A_{16}A_{18}A_{21}\ A_{22}A_{23}A_{24}A_{25}A_{26}A_{29}A_{31}A_{33}A_{35}A_{36}A_{38}A_{39}A_{40}\ A_{41}$
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

$A_{43}A_{44}\ A_{48}\ A_{49}\ A_{50}A_{52}\ A_{56}\ A_{57}\ A_{58}A_{60}A_{62}A_{63}A_{64}A_{66}A_{67}A_{68}A_{70}A_{72}A_{73}\ A_{75}A_{76}A_{77}\ A_{78}$
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48

$A_{83}A_{84}A_{85}A_{89}\ A_{91}A_{92}A_{93}A_{94}\ A_{95}A_{97}\ A_{98}\ A_{100}$
49 50 51 52 53 54 55 56 57 58 59 60

3.3. Аналогично выше изложенному выполняется 3-ий цикл отбора новой последовательности цифровых кодов (13 цифровых столбцов из 60) и далее продолжается отбор по 4-му циклу (после 3-его остается 60-13=47 столбцов).

В итоге после четырех циклов отбора остается 47-11=36 цифровых столбцов. Эти столбцы имеют много лакун (пропусков в своей последовательности), что позволяет завершить неочевидное построение новой последовательности цифровых кодов. Оставшиеся 36 столбцов могут быть распределены после (или же «до») столбцов, отобранных в ходе четырех циклов построения новой последовательности цифровых кодов. – Запишем их «после». Получаем новую последовательность 100 цифровых кодов:

A_1 A_5 A_7 A_{15} A_{20} $A_{27}A_{28}$ $A_{32}A_{34}$ $A_{42}A_{47}$ A_{54} A_{55} A_{59} $A_{61}A_{69}A_{74}A_{81}A_{82}A_{86}$ $A_{88}A_{96}$
$A_2A_{11}A_{12}A_{17}A_{19}A_{30}A_{37}A_{45}A_{46}A_{51}A_{53}A_{65}A_{71}A_{79}A_{80}A_{87}A_{90}A_{99}A_8A_{21}A_{22}A_{26}A_{31}A_{43}$
$A_{52}A_{64}A_{66}A_{72}A_{75}A_{91}A_{97}A_{10}A_{13}A_{23}A_{25}A_{41}A_{56}A_{68}A_{70}A_{78}A_{84}$ A_{100} $A_3A_4A_6A_9$ $A_{14}A_{16}$
A_{18} $A_{24}A_{29}A_{33}A_{35}A_{36}A_{38}A_{39}A_{40}A_{44}A_{48}A_{49}$ $A_{50}A_{57}$ A_{58} A_{60} $A_{62}A_{63}$ A_{67} $A_{73}A_{76}A_{77}$
A_{83} $A_{85}A_{89}A_{92}A_{94}A_{95}A_{98}A_{100}$

4. Над данной последовательностью цифровых кодов, построенной *неочевидным образом*, можно последовательно повторить 10 букв избранного алфавита и вновь закодировать *отрывок условного текста*. В итоге получим базу данных для кодирования защищаемой информации. Но при этом избранный в количестве 10 букв алфавит повторится 10 раз (10·10=100), а далее алфавит станет повторяться над повторенной записью цифровых кодов. Поэтому повторно встречающиеся в кодируемом тексте *символы* (буквы) получат только 10 новых цифровых кодов (например, буква **а**: A_1, A_{11}, A_{21} … A_{91} и далее вновь A_1, A_{11}, A_{21} …) и если кодируемый текст продолжается, то после получения новых 10 цифровых кодов повторяющиеся символы станут получать повторяющиеся цифровые коды, что позволит при *несанкционированном дешифровании* легко найти *ключ* к расшифровке закодированного текста по таблицам относительных частот встречаемости в тексте букв того или иного алфавита [1, Приложение 1.4].

Например, в русском тексте наиболее часто встречаются буквы **о (**доля числа букв на 1000 букв текста составляет 0, 090 т.е. около 90 на 1000 букв русского текста); **е** и **ё (**0, 072); **а** и **и** (по 0, 062) и т.д. до букв **э** и **ф** (0,003 и 0,002). Поэтому при 10 вариантах цифрового кодирования открытых символов *криптостойкость* текста будет низкой.

Целесообразно сделать количество различных вариантов цифрового кодирования каждого символа открытых данных равным количеству столбцов цифрового кодирования (т.е. при 2-х цифровом кодировании – 100 вариантов; при 3-х цифровом – 1000 и т.д.).

Количество вариантов кодирования каждого символа открытых данных станет равно количеству столбцов цифрового кодирования, если число символов кодирования и число столбцов для кодирования являются взаимно простыми (т.е. оба этих числа будут либо разными простыми > 2, либо составными числами, не имеющими общих множителей).

Например: если число кодируемых символов 9 (3·3), то при двух цифровом кодировании любой из 9 кодируемых символов может получить 100 цифровых кодов. Действительно: при 11-кратном повторе 9 символом над матрицей цифрового кодирования девятый символ получит код 99-го порядкового столбца матрицы цифрового кодирования (9+9·10). Мысленно повторяя последовательность цифровых кодов, получим вновь 100 её столбцов. Тогда, при последовательной **двенадцатой** записи символов кодируемого алфавита, 9-ый символ получит цифровой код 8-го порядкового столбца матрицы цифрового кодирования (99+9=108;108=100+8). А в первом варианте кодирования 9-ый символ получил код 9-го порядкового столбца. И т.д. При **сотом** повторе кодируемых символов (9·100=900) запись 100 столбцов цифрового кодирования повторится 9 раз. При этом 9-ый кодируемый символ будучи 100 раз повторно встретившийся в кодируемом тексте, получит цифровой код **сотого** цифрового столбца из новой последовательности цифровых кодов. А далее при сто первом, 102-ом и т.д. повторах алфавита кодируемых символов их цифровые коды станут повторяться: 101-ый одинаковый с 1-ым; 102-ой одинаковый со 2-ым и т.д. Т.е. по формуле идентичности: (100+k) = k, где k – число пробегающее значения от 1 до 99.

Руководствуясь изложенным введем правило: если количество кодируемых символов (алфавит…) является составным числом, имеющим общие множители с числом столбцов цифрового кодирования, то для получения количества вариантов кодирования любого символа, равного количеству цифровых столбцов, необходимо дополнить количество символов кодирования тем *минимально необходимым* количеством, при котором число символов кодирования станет взаимно простым с числом цифровых столбцов (т.е. эти числа не будут иметь общих множителей).

Добавочные символы кодирования могут либо иметь некое конкретное значение (например, знаки препинания или интервалы между словами), либо не выражать какой-либо

информации (такие символы можно назвать «пустые символы»). Но при этом: количество вариантов кодирования любого из символов изначально избранного их количества будет уменьшено на количество добавочных символов, т.е. количество добавочных кодируемых символов соответственно уменьшает количество возможных вариантов кодирования первоначально избранных символов открытых данных. Так, если, к принятому количеству букв русского алфавита -32 буквы, добавить 1 «пустой» символ: 32+1=33, то общее количество возможных вариантов кодирования станет 100 (вместо 25=100/4, где 4=2·2- общие множители чисел 32 и 100), но из них 1 вариант придется на «пустой»- 33-ий символ, поэтому любая из 32 букв алфавита может быть закодирована (100-1)= 99 вариантами.

4.1. Исходя из выше изложенного по избранному *условному* алфавиту из 10 русских букв (буквы от а до й - см. 2-ую стр. статьи) на основе новой последовательности цифровых кодов (см. п. 3.3) *зашифруем* предложение: «Ида, иди в джаз».

Для увеличения вариантов кодирования добавим 1 «пустой» символ к 10 буквам (символ обозначен прочерком). Беглый анализ текста показывает трехкратное повторение букв **и** и **д**, поэтому для кодирования повторно встречающихся букв без повтора цифровых кодов достаточно над последовательностью цифровых кодов 3 раза повторить алфавит:

а б в г д е ж з и й - а б в г д е ж з и й -

A_1 A_5 A_7 A_{15} A_{20} $A_{27}A_{28}$ $A_{32}A_{34}$ $A_{42}A_{47}$ A_{54} A_{55} A_{59} $A_{61}A_{69}A_{74}A_{81}A_{82}A_{86}$ $A_{88}A_{96}$

а б в г д е ж з и й - ……

$A_2A_{11}A_{12}A_{17}A_{19}A_{30}A_{37}A_{45}A_{46}A_{51}A_{53}$ ……

Первые 3 буквы – «ида» не повторяются, поэтому кодируются цифровыми кодами первого написания алфавита: A_{34} A_{20} A_1 или по номерам столбцов *исходной последовательности*: 331900. Далее буква **и** встречается повторно и получает код из 2-го написания алфавита- A_{86} . Следующая буква **д** также повторяется, но т.к. кодирование уже производится по другим цифровым кодам, то повтор алфавита не нужен и буква **д** получает код A_{69} (буквы **ид** кодируются: 8568). Далее 3-ий раз встречается буква **и**, что требует переход к кодированию по 3-ему повтору алфавита: A_{46} (45). Следующие буквы **в ж з** встречаются в тексте впервые; буква **д** встречается 3-ий раз (как **и)**, а буква **а** – во 2-ой , но т.к. переход на коды по 3-ему повтору алфавита уже произведен, то **и в д ж а з** кодируются по 3-ему повтору алфавита:$A_{46}A_{12}A_{19}A_{37}A_2A_{45}$ или по цифровым кодам: 451118360144. В целом кодируемое предложение: 3319008568451118360144.

Представим, что, стремясь к *незаконному использованию информационных ресурсов авторизованных субъектов* [3, с.11], *неавторизованный субъект*, дешифруя закодированное предложение, предположит, что символы открытых данных распределены последовательно над цифровыми кодами (а так и есть в примере!), тогда вопрос дешифрования значительно упрощается. Поэтому криптостойкость кодирования можно повысить путем неочевидного перемешивания *самих* кодируемых символов.

Предполагаемое количество информации, передаваемой в ходе информационного обмена, при х цифровом кодировании символов

Рассмотрим трех цифровое кодирование текста на русском языке. Перетасованная неочевидным образом последовательность цифровых кодов содержит 1000 столбцов. Алфавит из 32 букв дополним «пустым» тридцать третьим символом. При этом количество различных вариантов кодирования любого символа (буквы алфавита) составит 999 ≈ 1000.

Одна страница русского текста формата А5 содержит около 1600 знаков (букв). При этом буквы с наибольшей повторяемостью о, а, и, е и ё встречаются в тексте страницы: **о** - 1600·0,090=144 раза; **а** и **и** по 1600·0,062 = =99 раз ; е и ё суммарно - 1600·0,072=115 раз . При 1000 вариантах кодирования любого символа и, считая возможным двух кратное повторение каждого кода, можно закодировать текст (исходя из буквы **о**) объемом 2000:144 ≈ 13, 9 страниц. А при однократном применении каждого символа 7 страниц. Однако, в некоторых словах и фразах буквы распределены не соответственно **средней** частоте их повторения. Так,

предложение «Состав сошел с рельс» содержит 5 букв **с** и только 2 – **о**. Поэтому при кодировании предложения пятикратное применение буквы **с** заставляет применять 5 вариантов кодирования по пяти распределениям алфавита. Учитывая неритмичность повторения кодируемых символов, следует ввести понижающий коэффициент α, учитывающий неритмичность буквенного распределения. Экспертно примем $\alpha \approx 0,7$. Тогда количество страниц кодируемого текста при 2-х кратном повторе кодов составит $13,9 \cdot 0,7 \approx$ $\approx 9,5$ с. и около 5 с. при однократном повторе. При 4-х цифровом кодирование, объем кодируемой информации возрастет ~ в 10 раз (95 и 50 с.).

Декодирование зашифрованной информации

Имея единые алгоритм и соответствующую программу информационного обмена авторизованные абоненты на основе избранных числовых соотношений получают единые символы открытых данных и последовательность цифровых кодов.

Рассмотрим расшифровку закодированного сообщения на примере двух цифрового кодирования. Авторизованный пользователь имеет последовательность цифровых кодов (см. п. 3.3) и алфавит с добавочным 11-тым «пустым» символом. Для повышения криптостойкости перемешаем алфавит из 11 символов по соотношению 1/7=0, 142857, но применим обратный порядок цифр периода. В данном примере кодируемых символов мало, а все 6 цифр периода разные, поэтому перераспределение по ним букв алфавита можно упростить: а) перераспределение алфавита производится в обратном порядке цифр периода (от 6-ой цифры к 1-ой). Перед отбором для перераспределения символам открытых данных присваиваются последовательные номера от 1 до 11; б) согласно обратному порядку цифр периода дроби 7-ая буква – **ж** отбирается 1-ой буквой перераспределенного алфавита; 5-ая буква – **д** отбирается 2-ой; 8-ая (**з**) 3-ей; 2-ая (**б**) 4-ой; 4-ая (**г**) 5-ой; 1-ая (**а**) -6-ой. Оставшиеся символы, согласно присвоенным им номерам, можно распределить *перед* отобранными шестью буквами; в) в итоге алфавит примет вид: **в е и й – ж д з б г а** .

Зашифруем предложение: «Дай безе, Ида». – В предложении дважды встречаются буквы **д, а, е** . Буква **е** повторяется первой и при её повторе производится переход на кодирование по 2-ой последовательности алфавита. При этом повторно встречающиеся буквы **д** и **а** получают иные цифровые коды, чем в 1-ой последовательности алфавита, поэтому смена порядковой последовательности алфавита не требуется. – Итак, для кодирования данного предложения необходимо дважды повторить перераспределенный алфавит над ранее полученной последовательностью цифровых кодов:

в е и й - ж д з б г а в е и й - ж д з б г а

$A_1\ A_5\ A_7\ A_{15}\ A_{20}\ A_{27}A_{28}\ A_{32}A_{34}\ A_{42}A_{47}\ A_{54}\ A_{55}\ A_{59}\ A_{61}A_{69}A_{74}A_{81}A_{82}A_{86}\ A_{88}A_{96}$

Зашифрованная запись предложения: $A_{28}\ A_{47}\ A_{15}\ A_{34}\ A_5\ A_{32}\ A_{55}\ A_{59}\ A_{81}\ A_{96}$ или в записи по двух цифровым кодам: 27461433043154588095. А без перераспределения алфавита запись: 19004104263173856853.

У *санкционированного пользователя* построена та же последовательность из 100 цифровых столбцов. В базе данных компьютера для работы по заданному алгоритму декодирования имеется 100 цифровых наборов по 11 символам. Каждый цифровой набор соответствует *своему* порядковому распределению алфавита 11 символов. Абонент информационного обмена среди исходных данных (числовые соотношения, совокупность символов ...) получает код порядкового номера распределения кодируемых символов, с которого начинается кодирование сообщения, например, n=37. При декодировании один или несколько последовательных цифровых кодов соответствуют n-тому распределению кодируемых символов и эти коды по соответствующей программе «узнаются» и расшифровываются по n-тому распределению, но как только очередной цифровой код окажется принадлежащим n+1-ому (или n-1 и др.) распределению *кодируемых символов,* программа расшифрования «узнает» этот переход и переводит цифровой код в открытые данные через n+1-ое (n-1 или иное) распределение *кодируемых символов* и т.д.

Основы алгоритма информационного обмена

Детальное изложение информационного обмена методом числовых соотношений демонстрирует сложность и многоступенчатость составления алгоритма и соответствующей программы. Алгоритм представляет ряд отдельных операций (каждая со своим набором действий), объединенных целью: организация информационного обмена между санкционированными пользователями посредством шифров высокой криптостойкости.

Отдельные операции, состоящие из многих действий, имеют сугубо свое назначение (например, вычисление периода десятичной дроби) и объединяются в систему лишь в рамках названной выше *единой цели*, поэтому *общий* алгоритм представлен отдельными *блоками*.

Вследствие ограниченных возможностей детальной проработки *метода кодирования*, общий алгоритм построен на весьма упрощенном примере. При необходимости принятия решения *субъектом* (например, в ходе подготовки исходных данных, каких-либо сбоев и т.п.) предусматривается вывод результатов отдельных действий на экран.

Целесообразно выделить следующие блоки *общего алгоритма*:

Блок А – блок подготовки данных для последующего санкционированного информационного обмена. Блок включает следующие отдельные операции:

А1 - представляет набор последовательных действий для расчета, проверки и выбора периода десятичной дроби с целью последующего построения новой последовательности цифровых кодов:

1. задать соотношение **a/p**, где целые числа: **p** – простое и $\mathbf{1 \le a \le p-1}$;

2. вычислить период десятичной избранной дроби **a/p** до (p-1)-ой цифры.;

3.сверка цифровой последовательности периода дроби (p-1) с фактическим периодом:

а) если в количестве (p-1) цифр в фактической последовательности цифр нет цифровых повторов по двум интервалам длиной по (p-1)/2 цифр, либо четырем интервалам по (p-1)/4 цифр и т.п., то фактический период десятичной дроби совпадает с расчетным и автоматически принимается к перераспределению столбцов *исходной последовательности цифровых кодов*;

б) если в количестве (p-1) цифр в фактической последовательности цифр имеются цифровые повторы по двум интервалам длиной по (p-1)/2 цифр, либо по 4-рем интервалам по (p-1)/4 цифр и т.п., то фактический период соотношения в 2 либо в 4 и т.п. раз меньше расчетного (p-1). При этом на экран лицу, принимающему решение, выводится расчетная величина фактического периода заданного соотношения (пример: (13-1)/2=6 вместо 12).

б1) если фактический период достаточно велик, например, 96 цифр вместо 192, то субъект волен избрать период 96 цифр и дать сигнал к переходу на построение матрицы цифрового кодирования по данной цифровой цепочке;

б2) если же длина периода недостаточна, то задается иное соотношение a_2 / p_2 и производятся последовательные действия по п.п. 1-3.

А2 - количество цифр для кодирования 1 символа открытых данных передается без шифрования: 2 или 3… .

А3 – выбор способов шифрования для последующей передачи числового соотношения (соотношений);

А4 – между *абонентами санкционированного информационного обмена* должны быть обусловлены наборы символов *открытых данных* для информационного обмена. Так, к примеру, можно обозначить римскими числами изложенные в общепринятой последовательности алфавиты: русский- I, английский – II, латинский –III; цифры (0-9) –IV; знаки препинания – V и т.д.

Допустим *абоненту* переданы цифры I-IV, что означает указание: распределить в общепринятой последовательности алфавиты и цифры.

Если для повышения криптостойкости следует перемешать последовательность символов I-IV, то для этого при формировании *исходных данных* избирается некое числовое соотношение a/p (см. **А1**).

А5 – установить последовательность передачи подготовленных *исходных данных авторизованным пользователям*. Например, такую: 1) передается число цифр кодируемого символа, например, **2.** Далее ставится двойной интервал и 2) передаются римскими числами по **А4** символы открытых данных (каждое число отделяется одним интервалом);
3) если к числу символов открытых данных необходимо добавить «пустые» символы, то их количество, например, **1**, передается через 2 интервала от данных п. 2); 4) если необходимо перемешивать символы открытых данных, то через два интервала от данных по п. 3) передаются избранным методом соответствующее числовое соотношение а/р для расчета периода десятичной дроби, определяющего алгоритм перемешивания символов; 5) если шифрование предусматривается с n-того порядкового распределения совокупности кодируемых символов, то через два интервала от данных п. 4) сообщается порядковый номер, например, **5.**

Блок **Б** – передача *авторизованным пользователям* подготовленных по блоку **А** *исходных данных* .

Блок **В** – построение базы данных для кодирования защищаемой информации в виде неочевидной последовательности цифровых кодов и символов открытых данных и далее кодирование открытых данных.

Блок **Г** – информационный обмен между *авторизованными пользователями*.

Заключение

Предлагаемый метод кодирования состоит из семи эвристически определенных этапов, смысл и последовательность которых неочевидны.

Для распределения цифровых кодов и их применения при шифровании избираются цифровые последовательности десятичных периодических дробей. Один и тот же символ, повторно встречающийся в кодируемых данных, получает новый цифровой код. Количество различных цифровых кодов для кодирования любого повторно встречающегося символа определяется количеством цифр в цифровом коде. Множество вариантов перемен последовательности цифровых кодов *практически* неисчислимо, что позволяет обеспечить высокую криптостойкость шифра.

Периоды различных десятичных дробей *уникальны*, как уникальны числовые соотношения их образующие. Но числовые последовательности любого периода могут повторяться в виде интервалов тождественных цифровых последовательностей в существенно больших периодах иных числовых соотношений.

По мнению авторов, изложенный метод защиты информационного обмена от несанкционированного доступа оригинален, но требует более детального исследования и доработки с оценкой уровня криптостойкости при различных вариантах реализации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Нечаев В.И. Элементы криптографии (Основы теории защиты информации).- М.: Высш. шк., 1999. – 109с.

2. Защита информации в персональных ЭВМ /Спесивцев А.В., Вегнер В.А., Крутяков А.Ю.и др.–М., Радио и связь,1993.-192с.

3. Чмора А.Л. Современная прикладная криптография.- М.: Гелиос АРВ, 2001.-256 с.

4. Фергюсон Н., Шнайер Б. Практическая криптография.: Пер. с англ. - М.: Издательский дом «Вильямс», 2005.- 424с. : ил.

5. Исагулиев К.П. Справочник по криптологии / К.П.Исагулиев – Мн.: Новое знание, 2004.-237 с.

6. Патент на полезную модель № 82889. МПК (51) G 06 F 12/16. Устройство криптографической защиты информации. / А.Ю.Мухопад, Б.Н.Антошкин, Ю.Ф.Мухопад, БИ № 13, 2009.

7. Патент на полезную модель № 82890. МПК (51) G 06 F 12/16. Устройство криптографической защиты информации. / А.Ю.Мухопад, Б.Н.Антошкин, Ю.Ф.Мухопад, БИ № 13, 2009.

8. Патент на полезную модель № 82974. МПК (51) H 04 L 9/00. Устройство криптографической защиты информации. / А.Ю.Мухопад, Б.Н.Антошкин, Ю.Ф.Мухопад, БИ № 13, 2009.

9. Патент на изобретение № 2475838. МПК (51) G 06 F 21/00, H 04 L 0/00.

Устройство криптографической защиты информации./ А.Ю.Мухопад, Б.Н.Антошкин, Ю.Ф.Мухопад, БИ № 5, 2011.

10. Иванчишин В.Б. Решение проблемы простых близнецов. / информационные системы контроля и управления в промышленности и на транспорте.- Иркутск: ИрГУПС, 2013.- Вып. 22.- с.178-188.

11. Математический энциклопедический словарь. - М.: Сов. Энциклопедия, 1988-847с.

REFERENCES

1. Nechayev V.I. Cryptography elements (Bases of the theory of information security).-M.: Vyssh. шк., 1999. – 109 pages.

2. Information security in personal COMPUTERS / Spesivtsev A.V., Vegner V. A., Krutyakov A.Yu., etc. – M., Radio and communication, 1993. - 192 pages.

3. Chmora A.L. Modern applied cryptography. - M.: Helios of ARV, 2001.-256 pages.

4. Ferguson N., Schneier B. Practical cryptography.: The lane with English - M.: Williams publishing house, 2005. - 424 pages: silt.

5. Isaguliyev K.P. The reference book on cryptology / K.P. Isaguliyev – Mn.: New knowledge, 2004.-237 pages.

6. Patent for useful model No. 82889. MPK (51) G 06 F 12/16. Device of cryptographic information security. / A.Yu. Mukhopad, B.N. Antoshkin, Yu.F. Mukhopad, BI No. 13, 2009.

7. Patent for useful model No. 82890. MPK (51) G 06 F 12/16. Device of cryptographic information security. / A.Yu. Mukhopad, B.N. Antoshkin, Yu.F. Mukhopad, BI No. 13, 2009.

8. Patent for useful model No. 82974. MPK (51) H 04 L 9/00. Device of cryptographic information security. / A.Yu. Mukhopad, B.N. Antoshkin, Yu.F. Mukhopad, BI No. 13, 2009.

9. Patent for an invention No. 2475838. MPK (51) G 06 F 21/00, H 04 L 0/00.

Device of cryptographic information security. / A.Yu. Mukhopad, B.N. Antoshkin, Yu.F. Mukhopad, BI No. 5, 2011.

10. Ivanchishin V.B. Solution of the problem of ordinary twins. / information systems of control and management in the industry and on transport. - Irkutsk: ИрГУПС, 2013. - Issue 22. - page 178-188.

11. Mathematical encyclopedic dictionary. - M.: Sov. Encyclopedia, the 1988-847th.

Информация об авторах

Юрий Федорович Мухопад - д. т. н., профессор, профессор кафедры «Автоматизация производственных процессов», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: bts48@mail.ru

Виктор Борисович Иванчишин – научный сотрудник ООО «Байкальский научный центр», г. Иркутск, e-mail: vebirk@mail.ru

Authors

Yurij Fedorovich Muhopad – Dr. Sc., Professor, Professor of the Department «Automation of production processes»", Irkutsk State Transport University, Irkutsk, e-mail: bts48@mail.ru

Viktor Borisovich Ivanchishin - Researcher, Baikal Research Center LLC, Irkutsk, e-mail: vebirk@mail.ru

Для цитирования

Мухопад Ю.Ф., Иванчишин В.Б.. Кодирование защищаемых массивов информации на основе числовых соотношений . // «Информационные технологии и математическое моделирование в управлении сложными системами»: электрон. науч. журн. – 2018. – №1. – С. 72- – Режим доступа: http://ismm-irgups.ru/toma/11-2018, свободный. – Загл. с экрана. – Яз. рус., англ. (дата обращения: 01.10.2018)

For citation

Mukhopad Yu. F., Ivanchishin V. B. Kodirovanie zashchishchaemyh massivov informacii na osnove chislovyh sootnoshenij [Coding of the protected arrays of information on the basis of numerical ratios] // Informacionnye tehnologii i matematicheskoe modelirovanie v upravlenii slozhnymi sistemami: ehlektronnyj nauchnyj zhurnal [Information technology and mathematical modeling in the man-agement of complex systems: electronic scientific journal], 2018. No. 1. P. 72-83 [Accessed 01/10/18]-