УДК 681.3

Терминальные программы «цифровой» передачи и обработки данных, энергетическая и информационная эквивалентность

С. В. Кулешов

Приведены примеры применения программируемой технологии по А. Н. Колмогорову (ПТК). Рассмотрены аспекты эффективного (в смысле длины сообщения) представления данных и обеспечения их безопасной передачи. Приведено обоснование взаимной зависимости энергетической и информационной составляющих передаваемых данных.

The application examples of programmed technology on A. N. Kolmogorov (PTK) are considered. Also considered aspects of effective (in sense of size) data presentation and of provision of security for data transmissions. Validation of mutual dependence of energy and informative components of transmitted data is given.

Кулешов Сергей Викторович – канд. техн. наук, научн. сотрудник лаборатории автоматизации научных исследований Санкт-Петербургского института информатики РАН.

Область научных интересов: исследования в области обработки изображений и видеосигнала, виртуализация каналов передачи данных и информационных носителей, исследования и построение семиологических информационных систем.

Введение

Исходя из предложенного А. Н. Колмогоровым в 1960-х годах подхода к идентификации индивидуальных объектов «основные понятия теории информации должны и могут быть обоснованы без помощи обращения к теории вероятностей и так, что понятия «энтропия» и «количество информации» оказываются применимы к индивидуальным объектам».

Рассмотрим ряд примеров, показывающих отличительные особенности и достоинства программируемой технологии по А.Н. Колмогорову (ПТК), изложенной в [1].

Терминальные программы

Согласно равенству Шеннона [2] для последовательности из M символов

$$I = -MK\sum_{j=1}^{j=Z} p_j \ln p_j , \qquad (1)$$

где Z – число символов в алфавите, p_j – их априорные вероятности.

При использовании кодирования количество символов выходной последовательности является монотонной функцией от информации I. Пусть S – входная последовательность символов, m – их количество, P – программа, порождающая последовательность S; $l(P) = n$ – длина программы. Согласно терминологии [3], P есть терминальная программа, т. е. программа, не требующая входных данных и формирующая выходные данные – последовательность S.

Назовем «сложной» последовательностью S такую, что

$$n = l(P_{m+1}) = l(P_m) + 1 . \qquad (2)$$

Назовем «простой» последовательностью S такую, что

$$n = l(P_{m+1}) = l(P_m) . \qquad (3)$$

При этом $n \le m$, так как в самом худшем случае P просто выводит элементы S.

Введем следующий формальный язык для построения терминальных программ: оператор OUT x – вывод символа x; оператор $C(a, b)$ – повтор следующих a операторов b раз.

Каждый из операторов является единичным символом в алфавите P. Данный формальный язык является лишь одним из возможных и показывает особенности ПТК. Согласно классической теории об энтропийном кодировании, $l(P)$ должна быть неубывающей функцией при увеличении m (т. е. она не может быть сложнее «сложной» или проще «простой»).

Рассмотрим последовательность S следующего вида: S_{10}=ABBBBABBBB. Для ее кодирования терминальная программа P выглядит следующим образом ($l(P) = 4$):

C(3,2)
OUT A
C(1,4)
OUT B.

Для S_9=ABBBBABBB терминальная программа P выглядит следующим образом ($l(P) = 6$):

OUT A
C(1,4)
OUT B
OUT A
C(1,3)
OUT B.

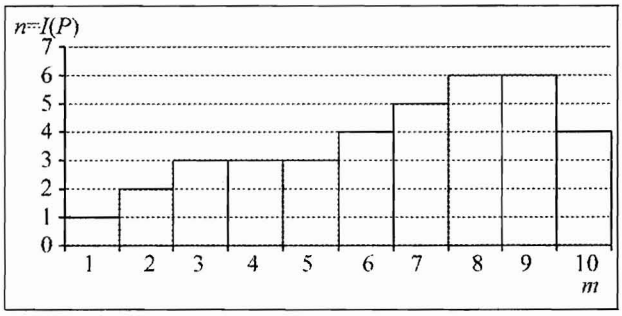

Рис. 1. Зависимость длины $l(P)$ от длины входной последовательности

Зависимость кратчайшей длины программы от длины входной последовательности S приведена на рис. 1. Тем не менее, согласно выражению (1) длина закодированного сообщения является неубывающей функцией от длины кодируемого сообщения.

Приведенный пример показывает, что количество информации по Шеннону не является критерием оценки сложности терминальной программы уникальной идентификации.

Количество информации отвечает лишь за энтропийную оценку ансамбля объектов, в то время как терминальная программа оценивает уникальные свойства самого объекта.

Информация и энергия

...«Рассмотрим основные понятия теории информации. Исходным будем считать понятие условной энтропии объекта x при заданном объекте y, $H(x|y)$, которую можно интерпретировать как количество информации, необходимое для задания объекта x в обстановке, когда объект y уже задан.

Обозначая через ϕ «заведомо заданный объект», получим безусловную энтропию $H(x|\phi) - H(x)$.

Информация, содержащаяся в объекте y относительно объекта x, определяется формально при помощи вычитания $I(x|y) = H(x) - H(x|y)$»... [4]

Рассмотрим переход от условной энтропии объекта к объектной идентификации, постулируемой Колмогоровым как минимальная длина, записанной в виде последовательности нулей и единиц «программы» P, которая позволяет построить объект x, имея в своем распоряжении объект y, т. е. $H(x|y) = \min l(P)$. Более подробно этот переход рассмотрен в [1].

Такой подход позволяет отождествить переход от энергетическо-энтропийного критерия к битовому представлению программы и подмене **данные-программа-данные**, которая изменяет концепцию передачи данных. При «цифровой» связи передаваться могут не только последовательности данных (т. е. битовое представление), но и непосредственно терминальные программы, порождающие эти данные также в их битовом представлении [3].

Рассмотрим физическую эквивалентность осуществления передачи импульсов как битового потока и энергией, требуемой для их передачи. При этом бит уже следует воспринимать и как информационную, и как физическую единицу.

Заметим, что в 1962 г. Дж. Пирс [5] предложил наглядную схему связи между битом и его энергией через производимую им работу. В соответствии с физическими законами для определения состояния системы при температуре T потребуется энергия

$$W = 0,693k\log_2 n = kT\ln n, \tag{4}$$

где W – энергия, k – постоянная Больцмана, n – количество возможных состояний, T – температура.

Минимальная мощность P передатчика при скорости потока S, бит/с

$$P = 0,693kTS. \tag{5}$$

Пример связи объектной и ансамблевой энергетической идентификации проявляется в телекоммуникационных системах, где слабым звеном является ограниченность объема передаваемых бит энергетической емкостью аккумулятора.

Возьмем для примера сотовый телефон. Общеизвестно, что его батареи аккумуляторов хватает на определенное время работы во время разговора.

На что расходуется эта энергия? Если не учитывать вспомогательные функции типа вывода на экран, подсветки и других непосредственно не относящихся к передаче голоса функций, то она расходуется на питание процессора, звуковых цепей и передатчика.

В связи с тем, что в телефонах формата GSM передача производится в цифровом формате, то время работы батареи зависит от числа переданных битов данных:

$$P = \frac{klS(I)}{t}, \tag{6}$$

где P – потребляемая мощность, Вт; l – длина информационного сообщения, бит; t – время передачи, с; $S(I)$ – функция сложности информационного сообщения, показывающая, во сколько раз можно компрессировать сообщение I; k – коэффициент, Вт·с/бит.

Присутствие функции $S(I)$ требуется из-за того, что передаются не все биты оцифрованного звукового сообщения, так как использование в канале передачи процессоров позволяет компрессировать входной поток. Значение $S(I)$ показывает насколько эффективно алгоритм компрессии, реализованный на процессоре, сжимает входные данные.

При этом необходимо учитывать, что для более эффективной компрессии в общем случае требуется более быстродействующий процессор, который в свою очередь потребляет большую мощность. Естественно, что потребляемая процессором мощность также зависит от его особенностей и технологии его производства.

Для современной технологии производства (2005 г.) и типовых параметров мобильного телефона (скорость GSM потока 13 кбит/с, емкость аккумулятора 800 мА/ч, время работы в режиме разговора 4 ч) можно определить, что для передачи одного бита данных требуется энергии порядка 0,06 мДж.

Это позволяет предложить следующее концептуальное соотношение бит-энергия:

$$I \le KE = Kmc^2, \tag{7}$$

где $E = mc^2$ – энергия, требуемая для передачи сообщения I бит, K – коэффициент Колмогорова, показывающий эффективность выбранной $l(P)$ программы: форматов и протоколов хранения и передачи данных, а также уровень развития технологии и КПД оборудования. Увеличение скорости процессоров и памяти за счет освоения новых физических элементов приближает K к теоретически возможному пределу формирования физически устойчивых состояний «1» и «0».

Вместе с этим известным фактом является то, что масса информационного носителя (например CD-R диск) одинакова до записи («чистый» диск) и после записи (диск с данными), ведь количество вещества не изменяется, изменяется лишь состояние отдельных его элементов. Приведенная формула показывает не изменение массы при наличии и отсутствии информации, а минимальную массу носителя для сохранения требуемого объема данных (задача хранения) или массивный эквивалент минимальной энергии для передачи требуемого объема данных (задача передачи данных).

Пусть требуется передать сообщение при скорости S, бит/с и для передачи одного бита данных требуется затратить энергию P, Дж (с учетом мощности передатчика и энергопотребления его вспомогательных узлов). Следовательно, для передачи указанного сообщения требуется мощность PS, Вт.

Использование алгоритмов компрессии позволяет уменьшать длину сообщения и, следовательно, использовать меньшую скорость передачи, что уменьшает требуемую мощность.

Вместе с тем, эффективность различных программ компрессии в среднем пропорциональна количеству операций над данными, что в сочетании с необходимостью выполнять передачу данных в реальном времени требует применения процессора соответствующей производительности. Производительность процессора, в свою очередь, также пропорциональна потребляемой им мощности.

Использование процессора для компрессии передаваемых данных способствует уменьшению скорости потока, что уменьшает энергопотребление передатчика. Одновременно с этим использование более эффективной компрессии увеличивает энергопотребление процессора, что может свести на нет выигрыш в энергопотреблении передатчика.

Из данных рисунков видно существование точки наиболее эффективной компрессии (и скорости передачи) данных с точки зрения энергопотребления. Такая точка может быть определена для конкретного типа данных, программ компрессии и используемой элементной базы, на которой реализованы передатчик и компрессор.

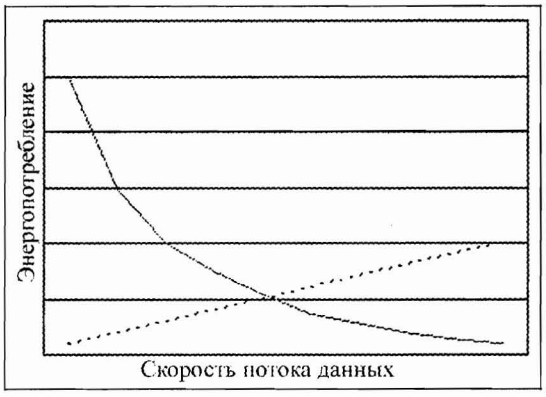

Рис. 2. Соотношение скорости передачи данных и энергопотребления: сплошная линия – энергопотребление передатчика без использования компрессии; пунктирная – энергопотребление с учетом компрессии

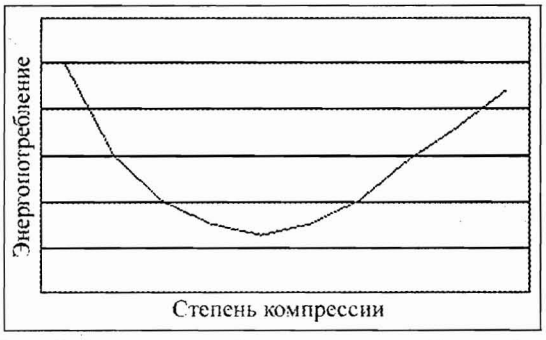

Рис. 3. Зависимость общего энергопотребления (передатчик + процессор) от степени компрессии данных

Естественно, что подобные выкладки не учитывают возможность радикального снижения энергопотребления, например, с помощью перехода на процессор, выполненный по другой технологии и обладающий теми же показателями производительности или разработки принципиально иных программ компрессии, а лишь указывают на существование точки энергетического оптимума и позволяют выбрать ее при заданных параметрах разрабатываемой системы.

Терминальные программы и информационная безопасность

Как показано в [1], любая процедура итерационно-рекурсивного представления обладает наилучшими возможностями минимизации программируемой избыточности. В качестве примера рассмотрим применение фрактальных представлений, для которых существуют уже разработанные программы их реализации [6]. Программируемая технология позволяет иначе взглянуть на проблему обеспечения безопасности передачи данных.

Ключевой проблемой технических средств защиты информации является появление действительно случайной последовательности бит. Дело в том, что генераторы случайных последовательностей, используемые для общих целей, являются псевдослучайными генераторами, так как в принципе существует конечное, а не бесконечное множество состояний ЭВМ, и, как бы сложно не формировалось в алгоритме число, оно все равно имеет относительно немного бит информационной насыщенности.

Рассматриваемый пример отличается от обычных методов шифрования тем, что фрактальная последовательность используется в качестве достаточно сложной кодирующей функции [7]. При этом описание этой функции, достаточное для построения, является набором вещественных чисел, которые задают начальные условия итерационного процесса построения фрактальной последовательно-

сти. Предлагаемый подход является вариантом гаммирования – процесса «наложения» гамма-последовательности на открытые данные, где в качестве гамма последовательности (последовательности псевдослучайных элементов) используется фрактальная последовательность.

Исходная концепция заключается в развитии идеи «почему днем не видно звезд на небе», т.е. в искусственном добавлении к исходным данным шумоподобного сигнала. Чтобы «увидеть звезды» нужно «выключить» «закрывающее» их солнце. Наиболее наглядно принцип работы данного метода можно проиллюстрировать на одноцветных растровых изображениях.

Пусть исходное растровое изображение представлено яркостными отсчетами, расположенными в матрице прямоугольного вида. Построим по начальным условиям итерационный фрактал. Затем, применяя некоторую обратимую функцию (например, поразрядную сумму по модулю 2), к парам значений точек исходного изображения и изображения полученного фрактала, получим новое изображение, которое и передается по каналу связи. Для расшифровки сообщения требуется, зная начальные значения процедуры построения фрактальной последовательности, восстановить изображение фрактала и, применяя операцию, обратную по отношению к операции передающей стороны (в рассматриваемом примере это также сумма по модулю 2), восстановить исходное изображение.

Принцип работы метода приведен на рис. 4. Как видно из схемы, сначала с помощью функции гаммирования вырабатывают гамма последовательность, которая зависит от параметра ключа K. После этого исходный сигнал просто суммируется с полученной гаммой по модулю.

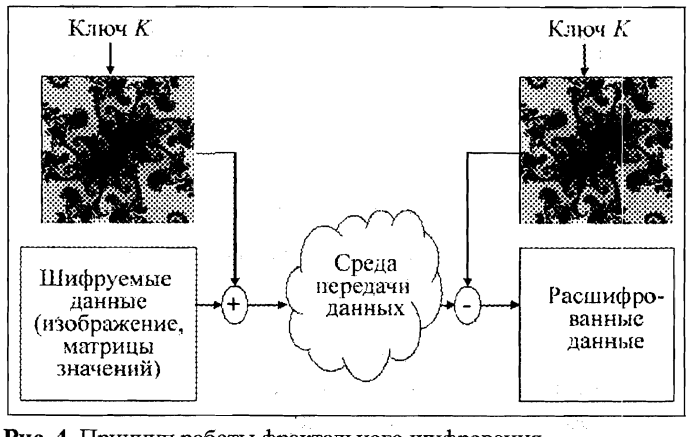

Рис. 4. Принцип работы фрактального шифрования

Начальные параметры итерационной функции, обеспечивающие выбор одного преобразования из совокупности возможных для данного алгоритма, являются криптографическим ключом.

Важно подчеркнуть, что если начальные значения итерационной функции построения фрактала взяты вблизи точки аттрактора, то требуется очень точное представление данных чисел, так как в этом случае фрактал обладает качественной неоднозначностью, что усложняет задачу подбора значений. При этом итерационная функция, порождающая фрактальную последовательность, является вычислительно необратимой функцией, т.е. легко вычислима в прямом направлении, в то время как определение значения ее аргумента при известном значении самой функции обладает сложностью, эквивалентной полному перебору. Иными словами, вычисление обратного преобразования не может быть произведено более эффективным способом, чем перебором по множеству возможных значений начальных параметров функции.

Итерационный процесс построения фрактала относительно долгий процесс, что при процедуре полного перебора начальных значений с последующим восстановлением фрактала и последующим расшифрованием требует больших вычислительно-временных ресурсов.

С точки зрения ПТК, основная идея фрактального подхода – в поиске минимальной сложности итерационного функционала, т. е. минимальной длины программы $l(P)$. Задача фрактального шифрования позволяет переформулировать задачу криптографии в терминах сложности по А. Н. Колмогорову [4], как построение минимальной программы, порождающей псевдослучайную последовательность при известных параметрах ключа и невозможность построения короткой программы восстановления последовательности при неизвестном ключе.

Рассматриваемый пример фрактального шифрования также демонстрирует возможность передачи дополнительной информации без увеличения объема передаваемых данных.

Отличительным свойством предлагаемого метода фрактального шифрования является проявление хаотических свойств фрактального сигнала (аналог большой длины криптографического ключа)

только при использовании начальных значений, определенных с высокой точностью, что требует предварительного составления каталога функций, удовлетворяющих заданным свойствам для эффективного выбора ключа для шифрования. При этом форма зависимости определяется видом конкретного итерационного функционала из каталога.

При оптимальном выборе начальных условий итерационного процесса и использовании асимметричного фрагмента фрактала можно обеспечить значимость всех битов выбранного ключа при выполнении операции шифрования. Полученный сигнал сложно отличить от шума, к тому же для расшифровки необходимо знать конкретный вид динамической системы и начальный параметр процесса.

Приведенный пример фрактального шифрования раскрывает иной концептуальный подход к организации процедуры как сокрытия, так и возможного дополнительного внедрения информационного содержания.

Выводы

Для цифровой программируемой технологии удалось получить концептуальное соотношение между объемом данных и требуемой для их обработки и передачи энергией. Данный эквивалент определяет пределы возможности цифровой полосы пропускания.

Характерной особенностью программируемой технологии является ее применимость для работы с данными любого типа. Так, канал, организованный для передачи битового потока, может использоваться как для передачи видео, так и для передачи текста.

Подобный подход реализует концепцию сверхширокополосной передачи данных.

Программируемая «цифровая» технология передачи данных реализует принцип подмены исходных данных терминальной программой, воспроизводящей семантико-смысловое содержание информационного сообщения на приемной стороне.

ЛИТЕРАТУРА

1. *Александров В. В., Кулешов С. В., Цветков О. В.* Концепция программируемой технологии «цифровой» теории связи: от герц к бит/с. – Информационно-измерительные и управляющие системы, 2007, № 6.
2. *Бриллюэн Л.* Наука и теория информации. – М.: Госиздфизматлит, 1960.
3. *Александров В. В., Кулешов С. В.* Нарротивные представления информационных процессов. – Информационные процессы, 2004, т. 4, № 2, с. 160–169.
4. *Колмогоров А. Н.* Три подхода к определению понятия «количество информации». – Проблемы передачи информации, 1965, № 1, с. 3–11.
5. *Пирс Дж.* Символы, сигналы, шумы. – М.: Мир, 1967.
6. Fractint Software – http://www.fractint.org/
7. *Кулешов С. В.* Фрактальное шифрование // Тр. СПИИРАН. Вып. 2, т. 1. – СПб.: СПИИРАН, 2004, с. 231–235.

Поступила 30 ноября 2006 г.