УДК 003.26.09

СПОСОБ СОЗДАНИЯ ДОПОЛНИТЕЛЬНОГО КАНАЛА СВЯЗИ, ОСНОВАННОГО НА ВРЕМЕННЫХ АЗБУКАХ МОРЗЕ

Р.Р. Нигматуллин

Казанский национальный исследовательский технический университет им. А.Н. Туполева - КАИ, кафедра радиоэлектроники и информационно-измерительной техники
Российская Федерация, 420110, г. Казань, ул. К. Маркса, 10

Аннотация. В этой работе рассматриваются новые методы кодирования, базирующиеся на множествах, основанных на азбуках Морзе. Эти множества временных азбук Морзе (ВАМ) могут быть успешно применены (по мнению автора статьи) в диапазоне КВ и УКВ. Более того, эта двоичная система (основанная на точках и тире) расширяет возможности существующей двоичной системы. К этой системе можно присоединить бесконечные псевдослучайные последовательности, полученные из простых чисел. Таким образом, три множества: (а) неисчислимое множество простых чисел; (б) бесконечные множества псевдослучайных чисел, полученных из простых чисел извлечением произвольных корней (также образующих множество натуральных чисел); и (в) присоединённые к ним ВАМ дают в итоге достаточно надежную систему, защищенную от взлома примененного кода, приближающуюся к "идеальному" решению. Заметим также, что предлагаемая система имеет удобные для передачи короткие крипто-ключи и может быть частично использована в существующих системах криптографии для защиты, в частности, выделенных информационных блоков.

Ключевые слова: временные азбуки Морзе; множества простых чисел**;** множества иррациональных и трансцендентных чисел; компактные крипто-ключи; множества дополнительных каналов связи.

1. Алфавиты, основанные на временных азбуках Морзе

Нет необходимости многократно повторять, что методы современной криптографии играют решающую роль в защите и передаче неискаженной информации от носителя закрытой информации к его потребителю. Основные методы криптографии основаны на анализе алгебраической модели [1-3]

$$SA = (X, K, Y, E, D). \quad (1)$$

Здесь X – множество истинных (открытых) сообщений; K-множество ключей; Y – множество зашифрованных сообщений; E-множество преобразований (операторов), которые преобразуют кодированное множество $Y = E(X)$; D-обратные множества, основанные на ключах, которые, будучи примененными к зашифрованным сообщениям, возвращают истинное сообщение, т.е. $X=D(Y)$. Дальнейшее развитие криптографии идет по линии совершенствования множеств K,Y,E,D. При этом предполагается, что алфавит, на котором формируется множество X, является неизменным или з*аданным*.

В этой работе упор делается на анализе криптографических свойств исходного алфавита, на котором строится множество сообщений X. Иными словами, ни в коем случае не претендуя на основы методов современной криптографии, можно добавить к ним также крипто-свойства исходных алфавитов, на которых строится множество сообщений. Эти зашифрованные наборы алфавитов усиливают криптостойкость самих сообщений и делают эти сообщения (передаваемыми по двум каналам) практически "идеальными". Чтобы увидеть эти скрытые свойства алфавитов, необходимо начать с анализа традиционной азбуки Морзе. Этот анализ приводит к тому, что на самом деле эти азбуки (а их можно построить значительное количество) можно сделать *временными* (т.е. явно зависящими от времени), что приводит к дополнительным возможностям по защите множества исходных сообщений X.

Итак, проблема, решаемая в этой работе, может быть *сформулирована* следующим образом: как соотнести точки и тире, присутствующие в азбуке Морзе (АМ), с бесконечными десятичными последовательностями, чтобы использовать их для построения множества алфавитов с целью усиления криптостойкости множества сообщений X? Анализ структуры исходной русскоязычной версии азбуки Морзе позволяет ответить на этот вопрос *положительно*.

Для понимания принципов кодирования алфавитов и как следствие построения их значительного количества рассмотрим детально (в качестве простейшего примера) традиционную азбуку Морзе (ТАМ), доступную любому пользователю и выложенную в Интернете. Если проанализировать русскоязычную версию ТАМ, то оказывается можно превратить её 60 символов (33 буквы русского алфавита, 10 цифр (0-9) и 17 (специальных знаков) в некую матрицу, состоящую из 60 строк и 6 столбцов. Она приведена в Приложении 1.

Для того, чтобы увидеть скрытые возможности ТАМ, проанализируем эту азбуку также с точки зрения длительности передаваемых сигналов. Можно выделить, как минимум, 6 временных интервалов различной длительности:

τ_0 – длительность передачи "точки";

τ_1 – длительность передачи символа тире;

τ_{00} – временной интервал между двумя последовательными точками;

τ_{01} – временной интервал длительности между точкой и тире;

τ_{10} – временной интервал длительности между тире и точкой;

τ_{11} – временной интервал длительности между двумя тире.

Для связи ТАМ с двоичной системой, примем, что "Точка" – это "нуль" в двоичной системе, а "Тире" – это единица. Учитывая особенности передачи АМ, положим, что нули перед единицами (длительность τ_0) и после них при связи заданной АМ с двоичной системой также имеют значение. Иными словами, особенность ТАМ состоит в том, что при преобразовании в двоичную систему нули перед единицей обычно *опускаются*, хотя в ТАМ они имеют *принципиальное* значение. Поэтому, чтобы учесть эту особенность, точки перед тире можно учесть указанием следующего символа: $c(z_n)$, $0 < z_n \leq 9$. Буква "c" - это *условная* цифра, которая соответствует *признаку* начальной точки (или нуля) в ТАМ. Можно взять в качестве этого признака *любое* другое число. Анализ ТАМ показывает, что вместо условной цифры "c", можно добавить ещё *девять* дополнительных цифр, отвечающих за *цвет* (раскраску) АМ. При необходимости впереди цифры z_n можно добавить следующие цифры, соответствующие 10 основным цветам: c_0- отсутствие цвета, c_1- красный цвет, c_2- оранжевый цвет, c_3- желтый цвет, c_4- зеленый цвет, c_5- голубой цвет, c_6-синий цвет, c_7- фиолетовый цвет, c_8- черный цвет, c_9 – белый цвет. Поэтому можно ввести ещё один параметр и составить комбинацию уже из двух передних цифр (c_nz_n), локализованных в интервале [0,99]. Если временные интервалы τ_{00}, τ_{01}, τ_{10}, τ_{11} между точками и тире оказываются более предпочтительными перед длительностями точки и тире τ_0, τ_1, определенными выше, то можно составить альтернативную АМ, основанную на промежуточных длительностях и использующую *только* интервалы между точками и тире. Эта интервальная АМ(τ) приведена в 6-ом столбце матрицы Приложения 1. Заметим, что она отличается от ТАМ, приведенной в 4-ом столбце, основанной целиком на длительностях τ_0, τ_1. Очевидно, что перед комбинацией (c_nz_n) можно добавить дополнительно электронную подпись (шифрованную комбинацию цифр), время передачи сообщения и персональные данные отправителя и другую зашифрованную информацию. Электронная подпись при блочном разбиении сообщений позволяет защитить истинное сообщение от её копирования и подмены [1-3].

Анализ ТАМ, представленный в виде матрицы, состоящей из 60 строк и 6 столбцов, показывает, что наиболее важной информационной составляющей является *третий* столбец. Этот столбец кодов АМ удобно представить в десятичном формате. Таблица ТАМ *не* учитывает заглавных букв. При необходимости можно ввести дополнительный 61-ый символ для признака *заглавной* буквы, отсутствующий в ТАМ. Чтобы не усложнять дальнейшее изложение остановимся на следующем обозначении $8(z_n)$, где восьмерка может означать возможность окраски, используемой АМ, в *черный* цвет. Структура ТАМ, представленная в Приложении 1, следующая: два первых последовательных столбца, расположенные слева, определяют структуру предлагаемого "словаря" $V(N)$ (цифра ↔ символ): n=1,2,…, (N= 60 = 33 + 10 + 17). Третий столбец, записанный в десятичной системе, определяет всю кодировку ТАМ и состоит из 6 цифр: первые 4 цифры + две цифры, представленные в виде десятичной дроби. Первые две целые цифры записываются в виде: $8(z_n)$, $0 < z_n \le 9$. Восьмерка (как уже было сказано выше) - это цифра, соответствующая "окраске" АМ в черный цвет. Восьмерку при необходимости и (для дополнительного кодирования) можно заменить и на другую цифру "раскраски" из интервала: $0 \le c_n. \le 9$. Следующие два целых десятичных числа (x_n,y_n) переводятся в двоичную систему, начиная с тире (единицы). Заметим, что при таком переводе в ТАМ, их двоичная комбинация не превышает величины 99 (1100011), что эквивалентно 7 двоичным разрядам, а точнее (в ТАМ) числу 56, соответствующему двоеточию (111000=46-(:)-[см. 46 строку второй колонки Таблицы 1]). Итак, если принять такое ограничение, то для этой пары чисел справедливо неравенство:

$$0 \le (x_n,y_n) \le 99. \qquad (2)$$

Простой комбинаторный расчет подсказывает, что без относительной привязки этих символов к русскому алфавиту, можно написать ровно $1000 = 10^3$ различных азбук Морзе (с учетом фиксированного символа (z_n), указывающего на число передних точек или нулей). Если учесть цифры "раскраски" $0 \le c_n. \le 9$, то число таких АМ возрастет до величины 10^4. Заметим также, что Таблица, представленная в Приложении 1, весьма гибкая. Пятый столбец предполагает, что АМ будет передаваться по *амплитудной* схеме с парой амплитуд (точка, тире) с заданными интенсивностями (превышающими уровень шума) и длительностями передачи точки (τ_0) и тире (τ_1), соответственно. Если противник ставит шумовую "завесу", то тогда с помощью фильтрации шума можно добиться получения заданных длительностей τ_{00}, τ_{01}, τ_{10}, τ_{11} и перейти на передачу информации уже по безамплитудной схеме, опираясь на комбинации этих длительностей, указанных в шестой колонке матрицы.

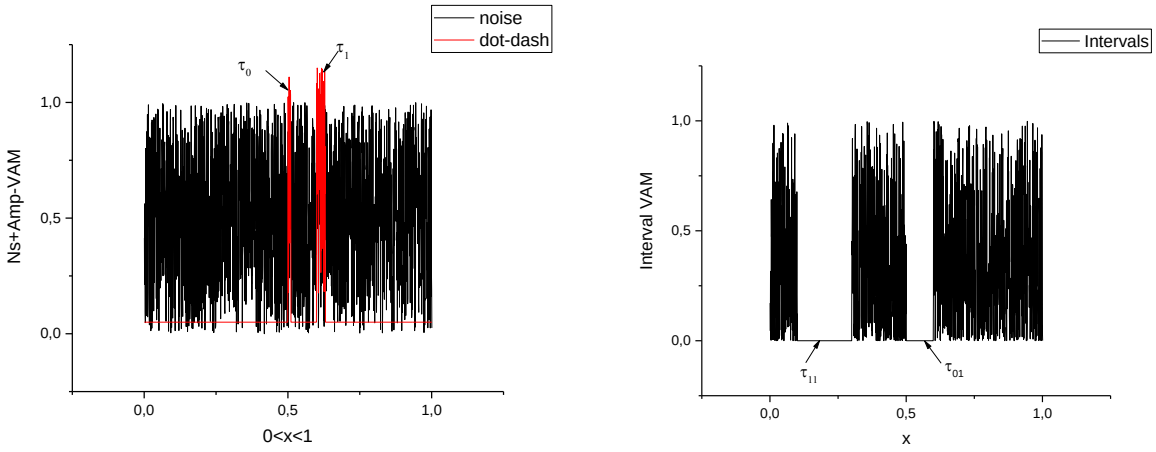

Рис.1. Этот рисунок поясняет различие между амплитудной (слева) и интервальной (справа) АМ

Следующие два числа, отделенные запятой (a_n,b_n), играют важную роль: эта пара обеспечивает "привязку" или соответствие буквам русского алфавита, т.е. группе чисел (1-33), числам (34-43) и специальным символам (44-60), указанным в Таблице 1

$$1 \leq (a_n,b_n) \leq 60. \quad (3)$$

В итоге мы имеем следующую структуру ТАМ из 6-и цифр в десятичной кодировке, представленную в третьем столбце:

$$S=8(z_n)(x_n,y_n)_2,(a_n,b_n). \quad (4)$$

Числа (a_n,b_n) для простоты записи в выражении (3) можно отделить запятой.

Максимальное число разрядов, которое занимает это число в двоичной системе можно оценить из соотношения $S(99)_2(99)_2(60)\rightarrow 20$ разрядов или бит (запятая между целыми и дробными частями опущена). С учетом запятой АМ (символ под номером 45) максимальное число разрядов увеличится до 26 бит. Для интервальной АМ (столбец 6) можно ввести специальную длительность (τ_P), отделяющую целую часть АМ от её дробной части. Тройка чисел $(z_n)(x_n,y_n)_2$, переведенная из 10-ой кодировки в двоичную, показана в пятом столбце матрицы ТАМ. Заметим, что цифра 60 также условная цифра. Можно добавить сюда заглавные буквы, "сленговые" выражения, служебные слова и буквы английского алфавита, просто добавляя к символу раскраски c_n дополнительный символ слева. Такое расширение ТАМ уже до 99 символов вместо заданных 60, должно быть соотнесено с *возможностями* аппаратуры, реализующей эту систему. Разумеется, число символов алфавита может быть неограниченным и вырасти до 10^s, где выделенная комбинация уже превышает возможности неравенства (2) $10^2 \leq (x_1,x_2,\ldots,x_s)_n \leq 10^s$ и занимает уже s разрядов.

2. Алгоритм и криптостойкость ВАМ

Внимательный анализ ТАМ, приведенной в приложении, показывает, что можно создать Временную Азбуку Морзе (ВАМ). К этой ВАМ можно дополнительно применить операторы кодирования E для преобразования его в зашифрованное сообщение ($y_k=E_k(x)$).

Сколько таких ВАМ можно создать? Несложный подсчет показывает, что *минимальное* число таких ВАМ, обеспечивающих некое соответствие 60 символов второго столбца матрицы дробной части десятичных чисел, расположенных в третьем столбце, можно оценить с помощью формулы:

$$Q = (10^4)\cdot 60 = 6\cdot 10^5. \quad (5)$$

Это число в силу его достаточно большой величины затрудняет раскодировку временного сообщения для третьей стороны. Максимальное число Q можно получить, если получать последнюю комбинацию пары чисел заменой $(60)\rightarrow(60!)$. Заметим, что это число (Q) определяет полное число азбук Морзе, которые можно получить на *конечном* этапе. Как можно усилить криптостойкость ВАМ и довести её до значительных величин? Откуда взять бесконечные последовательности для создания соответствия между третьим столбцом матрицы и некой псевдослучайной последовательностью (ПСП)? Для этой цели автор предлагает использовать бесконечный набор *простых* чисел и рекурсивные алгоритмы по созданию бесконечной последовательности из выбранного простого числа. Таким образом, в распоряжении пользователя будут две бесконечные последовательности (два дискретных множества с бесконечной мощностью): (а) таблица простых чисел (ТПЧ) и (б) алгоритмы, построенные на их основе. Применяя выбранный заранее алгоритм (эти пронумерованные алгоритмы предварительно заносятся в таблицу алгоритмов (ТА), и эта информация из ТА является *закрытой* для третьей стороны) для простых чисел, выбранных из ТПЧ, можно

создать последовательности сколь угодной длины из выбранного простого числа или их комбинаций. Чтобы эта ТА не занимала значительные объёмы памяти, эти алгоритмы нужно заменить *рекурсивными* алгоритмами, которые можно использовать в качестве *ключей,* открывающих доступ к ТПЧ и ТА. В этом случае, эти таблицы можно сделать открытыми.

Эти рекурсивные алгоритмы позволяют сгенерировать необходимые псевдослучайные последовательности (ПСП) данной длины. Как связать ВАМ с ТПЧ и рекуррентным алгоритмом, который создает необходимую ПСП? Именно ПСП (т.е. множество натуральных чисел, имеющее бесконечную мощность) заданной длины N_b и с фиксированной меткой k служит некой платформой для "привязки" русских символов, образуя заданную ВАМ. Метка k играет роль начального номера для отбора 180 чисел ПСП, пригодной для создания ВАМ. Чтобы противник не узнал ключевые цифры: PN_m (простое число m), №$A(PN_m)$ (номер рекурсивного алгоритма), (N_m, k_m) – (длину ПСП - N_m и метку - k_m , с которой начнется построение ВАМ) , их можно передать по открытому/закрытому каналу с помощью исходного ключа шифрования (КШ) и отобрать из него (уже по специальному алгоритму) нужные цифры.

Пример: Пусть Ответчик получил от Запросчика следующую цифровую колонку, переданную с помощью короткого ключа шифрования (КШ): 98765[43](7)[(4321)5]78228235360…. [5].

Ответчик понимает эти цифры так: начиная с последней цифры этой последовательности [5] нужно отсчитать пятую цифру, взять простое двузначное число (или ближайшее к нему слева в ряду чисел, если это число составное). В данном примере получилось число 43 – простое. Извлечь из него корень седьмой степени (7), отсчитать длину последовательности (N_m = 4321) и взять метку k_m =5. Понятно, что этот простейший способ шифрования взят в качестве примера. Можно придумать более изощренные способы по извлечению нужных простых чисел, необходимых алгоритмов и ПСП в посланном КШ Ответчику.

Для создания нужной ВАМ необходимо применить следующий алгоритм:

А(1). Выбрать по заданным координатам из ТПЧ простое число (PN_m) или их комбинацию (при практической реализации этой системы можно будет разработать оптимальный критерий выбора простого числа или подходящей пары из ТПЧ). Согласно теории вероятностей, самый лучший выбор (PN_m) из ТПЧ должен соответствовать их *равномерному* распределению.

А(2). Применить к нему алгоритм под заданным номером из ТА [№{$A(PN_m)$}].

А(3). Создать с помощью А(2) необходимую последовательность и выбрать на сгенерированной ПСП заданную длину (N_b) и метку (k).

А(4). Набор трехзначных чисел $(z_n)(x_n,y_n)_2$ при дополнительно добавленной фиксированной "цветовой" метке c_n на полученной ПСП и определенных на отрезке [k, k+180], будет определять структуру ВАМ для сообщений между двумя сторонами.

А (5). Последний этап – эта привязка тройки чисел к комбинации символов (a_nb_n). Для этого нужно исходную возрастающую последовательность $n(s_1,s_2,s_3)$ первой колонки матрицы "перетасовать" в случайном (для противника) порядке с помощью тройки чисел $\mathbf{s}(s_1,s_2,s_3)$. Эта перетасовка производится среди букв (33; $1\le s_1\le 33$), чисел (10; $1\le s_2\le 10$) и символов (17; $1\le s_3\le 17$). Причем эту перетасовку чисел $\mathbf{s}(s_1,s_2,s_3)$ можно поставить в зависимость от "цвета" (c_n) АМ. После этого создается заданная ВАМ по коду, приведенная в 3-ей колонке, соответствующая выражению (4).

Дополнительный этап. Если есть необходимость использовать интервальную ВАМ(τ), то для такого перехода необходимо задать заранее 7 интервалов: τ_0, τ_1, τ_{00}, τ_{01}, τ_{10}, τ_{11}, τ_P и перейти к кодировке, указанной в 6-ом столбце матрицы Приложения 1.

Поясним эту идею на простом примере. Как сделать выбор пары простых чисел из таблицы, практически недоступной 3-ей стороне? *Допустим*, что число простых чисел в ТПЧ равно 5133 (оно соответствует числу ПЧ, расположенных в натуральном ряду чисел [1-5·10^4]). Образуем из этих чисел матрицу 513×10, где простые числа располагаются по строкам слева направо и сверху вниз и приближенно соответствуют равномерному распределению. Тогда если в течение года (для оценки возьмем 350 дней) использовать ежедневно пару чисел из некоторого столбца этой матрицы, то число комбинаций за 10 лет эксплуатации такой таблицы из 5130 простых чисел достигает гигантской величины равной $\left(C_{513}^{2}\right)^{10} \cong 1.32\cdot10^{50}$! Заметим также, что к группе простых чисел можно добавить также и трансцендентные числа такие как Pi = 3.1415926535897932…, постоянную Эйлера E=2.7182818284590452… и другие. Итак, пусть из этой матрицы (N=513×M=10) выбраны два простых числа: $PN_{1,m}$. Тогда самые простые алгоритмы по созданию из них бесконечных последовательностей можно записать в виде:

$$A\left(PN_1, PN_m\right) = \left|\left(PN_1\right)^{1/2} \pm \left(PN_m\right)^{1/3} \pm \left(PN_1\right)^{2/3} \pm \left(PN_m\right)^{3/5}\right|. \qquad (6)$$

Этот алгоритм нетрудно превратить в зашифрованную последовательность, если условиться натуральные числа по извлечению корней располагать в порядке их возрастания. В данном примере эти числа такие: 1,m,1m;(+) 1123;2335. Число m соответствует номеру Ответчика. Модуль в выражении (6) нужен для того, чтобы иметь дело *только* с положительными последовательностями. Но если условиться принять для нулей *отрицательные* амплитуды, то можно отказаться от знака модуля в выражении (6). Понятно, что таких алгоритмов может быть значительное количество, так как последовательность простых чисел (как известно из современной теории чисел) *не* имеет функциональной зависимости от его номера в последовательном ряду натуральных чисел ($PN(n)$=? (n=1,2,…N,…). Их можно задать только *табличным* способом, что и было показано выше на простом примере. Несложный подсчет показывает, что даже эта простая формула (5) приводит к 2^3=8 различным бесконечным ПСП. Итак, допустим, что рекурсивная формула, примененная к выражению (6), дала заданную фиксированную "ленту" чисел, которая была обрезана до необходимой величины N_b с фиксированной меткой k (выбор конечного N_b и метки k происходит по договоренности 2-х "переговаривающихся" сторон) и может быть передана открытым способом с помощью зашифрованного сообщения или по закрытому алгоритму через КШ. Для передачи четырех параметров по КШ {(PN_m), [№{A(PN_m)}], (N_m,k_m)} - (простое число из ТПЧ, номер алгоритма из ТА, длина последовательности N_m и номер метки k_m) от Запросчика к Ответчику требуется минимальное время.

После применения алгоритма (А(1)-А(5)) получается необходимая тройка чисел (c_n=8)[z_n(x_n,y_n)] и привязка к символам (a_n,b_n) через цветовой параметр c_n. Отметим, что максимальный десятичный код АМ, выбранной по договоренности для некой ВАМ, выглядит так:

8[(9)(99)$_2$], (a_n,b_n) = ● ● ● ● ● ● ● ● ● – – ● ● ● – – , (a_n,b_n).

А минимальный код:

8[(0)(01)],(a_n,b_n) – , что соответствует простому тире АМ.

Нетрудно увидеть, что выбор числа 8[0(00)] *запрещен* в силу его неопределенности. Заметим также, что в приведенной Таблице ТАМ (Приложение 1), существуют две независимые точки и два тире. Пара точек и тире соответствует символам (7 и 20) ТАМ, а другие два символа соответствуют грамматическим символам ((44) и (52)), соответственно. Выбор однозначных символов происходит по соглашению между двумя сторонами. В этой части,

связанной с раскодировкой предлагаемой системы, приведем следующие оценки. Что нужно перебрать противнику для возможного раскодирования системы? Сколько переборов ему нужно сделать, имея самый быстрый компьютер?

1) Перебор ТПЧ. Возьмем для оценок тот же самый набор простых чисел равный PN=5133, который распределен в натуральном ряду чисел от [1-50000]. Если собрать эти числа в виде матрицы 513×10 и выбирать из неё (желательно равновероятным способом) два простых числа из одного столбца длиной L=513, то нужно перебрать число сочетаний $K_1 = [C_L^2]^{10} \cong 1.32 \cdot 10^{50}$ или взять полное число комбинаций из числа $[C_{PN}^2]$. Даже если один перебор занимает одну миллисекунду, то весь перебор займет примерно 10^{47} сек $\cong 0.3 \cdot 10^{40}$ лет! (для сравнения время существования Вселенной $1.3 \cdot 10^{10}$ лет).

2) Перебор алгоритмов. Пусть число алгоритмов, связанных с парой простых чисел будет равно, для примера, K=100. Предположим, что они будут вычисляться по формуле, вида:

$$A_k(PN_1, PN_2) = \left| \frac{1}{k} \sum_{p=1}^{k} \left(PN_1\right)^{p/K} \pm \frac{1}{k} \sum_{p=1}^{k} \left(PN_1 \cdot PN_2\right)^{p/K} \right|, \quad (7)$$

$$k = 1, 2, ..., K.$$

Очевидно, что формула (7) взята лишь в качестве примера. Формулы для расчета алгоритмов диктуются быстротой и надежностью реализации и выбираются по неким критериям заранее. Из этих K алгоритмов для переговоров (здесь и сейчас!) нужно взять всего лишь пару ПСП, каждая из которых будет иметь "шумовую ленту", состоящую из N_m (m – возможный номер Ответчика) десятичных цифр. Одна ПСП подойдёт для ВАМ (1), а другая ПСП для Ответчика ВАМ (m). Тогда число возможных комбинаций будет задаваться формулой $K_2 = C_K^2 \cong 1.2 \cdot 10^3$.

3) Следующий этап – это выбор отрезков длиной по 180 десятичных цифр (образующих заданный алфавит АМ) из каждой ПСП. Эта процедура соответствует фиксации параметра k на заданной ПСП длины N. Число таких комбинаций при фиксированной метке k будет равно $K_3=(C_N^{180})$. Это число нужно помножить на число сочетаний, определяющих выбор пары k и k +180 на заданной длине последовательности N. Число таких пар будет равно $K_4=C_P^2$ (Здесь P=[N/180] – целая часть отношения). Итак, только для выбора правильной пары ВАМ(1,m) потенциальному крипто-аналитику потребуется перебрать такое гигантское число

$$G = K_1 \cdot K_2 \cdot \left(K_3\right)^2 \left(K_4\right)^2. \quad (8)$$

Но на этом его проблемы по перебору астрономического числа комбинаций не заканчиваются. Необходимо число G умножить на (Qm = ($s_1(c_n) \cdot s_2(c_n) \cdot s_3(c_n)$), где ($s_1 \in [1,33]$, $s_2 \in [1,10]$, $s_3 \in [1,17]$) комбинаций, обеспечивающих соответствие ВАМ(1,m) символам русского алфавита. Эти числа (скрытно для третьей стороны) выбираются в зависимости от цветового параметра c_n. Если речь идет об интервальной ВАМ($\tau(1),\tau(m)$), то нужно перебрать комбинации интервалов, а эти интервалы для Запросчика $\tau(1)$ и Ответчика $\tau(m)$ будут у каждого свои.

Автор полагает, что приведенные выше оценки должны убедить потенциального криптоаналитика в бессмысленности взлома предлагаемой системы, основанного на переборе вариантов, даже имея самый мощный компьютер. Поэтому имеет смысл определить предлагаемую системе как "ВАМПИР". (Временная Азбука Морзе Почти "Идеальное" Решение). Заметим, что система ВАМПИР – весьма гибкая структура. Если даже один из её

вариантов по каким-либо неожиданным причинам становится известным противнику, то систему ВАМПИР(1) можно быстро поменять на другую ВАМПИР(2), просто "перетасовав" простые числа из ТПЧ(1) или перейти на другую ТПЧ(2), изменив также интервалы в шестом столбце. Поэтому степень защищенности (криптостойкость) этой системы весьма высокая.

Можно также выделить принципиальные особенности предлагаемой системы ВАМПИР при её сравнении с существующими криптосистемами:[1]

1) Гибкость системы ВАМПИР основана на трех бесконечностях: (а) бесконечной системы простых чисел Pr(n), (б) бесконечной системы натуральных чисел (n) и (в) бесконечной системы $\left[\Pr(n)\right]^{1/k}$ корней, извлекаемых из множества (а) с помощью множества (б). Эти три бесконечные последовательности придают гибкость этой системе и усиливают её криптостойкость. Заметим также, что предварительная проверка показывает, что распределение чисел в бесконечном множестве $\left[\Pr(n)\right]^{1/k}$ при достаточно больших выборках (N>>1) *равномерное*, что также важно при их проверке на соответствие критериям NIST.

2) Важно подчеркнуть также её *временный* характер. Допустим, что она по каким-либо причинам взломана. Эта система по договоренности с переговаривающимися сторонами может быть использована как инфо-оболочка в течение года, месяца, дня или даже часа. И каждый раз для переговоров будет использоваться свой алфавит. Совершенно очевидно, что ТАМ взята в качестве *примера*. Можно заранее договориться об использовании *любой* комбинации из букв и цифр, взятых из различных алфавитов (латинский, греческий и др.), которые также будут преобразовываться в двоичную систему.

3) Заметим также, что преобразование десятичной комбинации (используемой для понимания) в двоичные коды происходит специфически: целая и дробные части в выражении (4) преобразуются *независимо* друг от друга. Эта особенность также усиливает криптостойкость системы ВАМПИР.

4) Анализ существующих крипто-методов показывает, что большинство из них основано на формуле (1), где исходный алфавит предполагается постоянным. Предлагаемая система ВАМПИР указывает на новые возможности, возникающие в случае, если исходный алфавит – величина переменная.

5) Совершенно очевидно, что к полученной комбинации нулей и единиц из выбранного алфавита, можно добавить существующие методы шифрования, основанные на кодах Фейстеля или их обобщениях.

3. Схема взаимодействия "З" ↔"О" и возможная привязка к БПЛА

Сформированная группа "случайных чисел" (см. формулу (4)) образует некий зашифрованный алфавит, который может уже послужить дополнительной основой для шифрования традиционными методами. Эта ПСП передается от "З" к "О" по 2-м открытым каналам: (а) по первому каналу на выбранный сервер или спутник по двоичной системе (колонка 5) и (б) независимо по второму каналу, соответствующему образованному ВАМ(1). Известно, что для защиты от помех передаваемое сообщение в двоичных кодах должно быть *избыточным* [4]. Предполагается, что это условие выполнено.

Примем во внимание также тот факт, что комбинация "точек-тире" при передаче наиболее устойчива к атмосферным помехам и эту комбинацию можно передавать без заметных искажений даже в диапазоне коротких (1-30 МГЦ) и ультракоротких (30 МГЦ - 3 ГГЦ) волн и выше. Согласно литературным источникам, диапазон КВ вновь становится очень *популярным*, так как диапазон (1-30) МГЦ практически не поддается маломощным

[1] Особенности системы были сформулированы после обсуждения работы с рецензентами

средствам глушения с помощью средств РЭБ в силу специфики распространения коротких волн в атмосфере. Для третьей стороны эта передаваемая по открытому каналу ПСП выглядит как некий "шум". Если противник попытается исказить передаваемый сигнал, передаваемый по амплитудной схеме τ_0, τ_1, τ_P, то можно задублировать сообщение или перейти на *интервальную* передачу сигналов АМ, по схеме, указанной в столбце 6. Заметим также, что вместо ЭМ импульсов можно использовать также и световые, делая ВАМ поистине универсальной.

Как, примерно, организована передача ВАМ(1)↔ВАМ(m)?
Организация информационного сообщения или посыла начинается с ТПЧ, которая бесконечна в принципе. Но из неё всегда можно по договоренности обеих сторон отобрать конечную комбинацию. Применяя алгоритм А(1)-А(5), описанный выше, получаем необходимые комбинации, представленные формулой (3). С помощью этого алгоритма можно создать две АМ: ВАМ(1), относящейся к "З" и ВАМ(m) – к "О" (БПЛА). На сформированных ВАМ формируется ключ шифрования КШ($L(N_m)$, $l_m(1)$) заданной длины $l_m(1) < N_m$ для Ответчика. Повторим ещё раз, что передаётся *не* сама ПСП, а параметры (N_m, k_m) для воспроизведения ПСП по соответствующему алгоритму. Для простоты в круглых скобках мы опустили номер используемого алгоритма [№($A_m(PN_m)$]; они занесены предварительно в ТА и "зашиты" в передаваемом КШ. В самом начале обмена З↔О можно просто обменяться краткими паролями, созданными на ВАМ(1). Чтобы противник не повторил кодовую комбинацию, запрашиваемый Ответчик может ответить на той же ВАМ(1), но используя другую метку k_m. У каждого Ответчика своя метка и Запросчик знает "Своих" по их меткам k_m. После идентификации соответствующей метки, при необходимости диалог З↔О может быть продолжен и стать более *содержательным*, когда Ответчик переходит на свою ВАМ(m), выбрав последовательно (или по определенному правилу) 180 чисел по своей метке [k_m, k_m+180]. Таким образом, для Ответчика на число PN_1 созданы два канала для ответа на запрос: КШ(N_1, $l_1(1)$, k_1; $\boldsymbol{N_m}$, $\boldsymbol{l_m(1)}$, $\boldsymbol{k_m}$) по первому каналу, а по 2-ому независимому каналу передаются коды (выделены жирным цветом) для создания своего "языка общения" Ответчика ВАМ(m) с Запросчиком. После подтверждения правильности отзыва О→З, Ответчик получает право принять сигнал по КШ(N_1, $l_1(1)$, k_1; $\boldsymbol{N_m}$, $\boldsymbol{l_m(1)}$, $\boldsymbol{k_m}$) и сообщение по ВАМ(1) на создание своей ВАМ(m). В силу встроенной ТПЧ, он может поменять длину ленты $\boldsymbol{N_m}$ или поменять простое число $PN_1 \leftrightarrow PN_m$ (в зависимости от цели и особенностей ответа и цены БПЛА). Рассмотрим более простой вариант: Ответчик использует то же простое число PN_1. Тогда он формирует свой ответ на той же ленте чисел, меняя лишь её длину ($N_b \rightarrow N_m$) и метку ($k_b \rightarrow k_m$). В результате формируется *асимметричный* ответный ключ шифрования КШ($\boldsymbol{N_m}$, $\boldsymbol{l_m(1)}$, $\boldsymbol{k_m}$) с выбором другой привязки к азбуке символов (a_n',b_n'). ВАМ(m) будет иметь ту же структуру, что и ВАМ(1).

$$\text{ВАМ}(m):\ c_m\left[\left(z_n'\right)\left(x_n'y_n'\right),\left(a_n'b_n'\right)\right]. \qquad (9)$$

В азбуке ВАМ(m) заменён "цвет" начальной точки ("8" → "c_m"), остальная структура остается прежней. Смена цвета, как уже было отмечено выше, приводит к смене комбинации (a_n,b_n) → (a_n',b_n'). Переход от прежнего набора (a_n,b_n) к новому набору (a_n',b_n') происходит по специальному алгоритму. Запросчик принимает КШ($\boldsymbol{N_m}$, $\boldsymbol{l_m(1)}$, $\boldsymbol{k_m}$) от Ответчика, переводит его обратно ($N_m \rightarrow N_b$), ($k_m \rightarrow k$), проверяет *дополнительно* совпадение ключевых символов и окончательно подтверждает Опросник о (не)совпадении своим КШ(N_b, l_b(1'), k_b). Причем при тех же настройках можно поменять длину $l_b(1') \neq l_b(1)$. Кроме того, к услугам диалога З↔О всегда имеется другая ВАМ($\tau(1)$, $\tau(m)$), основанная на заданных интервалах. Её применение целесообразно, если противник пытается заглушить или исказить сигналы от первого варианта обмена, основанного на заданных амплитудах точки τ_0 и тире τ_1.

Понятно, что Ответчик может по второму каналу "переговорить" с Запросчиком, вновь используя свой ВАМ(m) как и при первом контакте. Таким образом, переговоры З↔О могут осуществляться по открытым каналам с каждым выделенным Ответчиком(m) по отдельности в широком диапазоне ЭМ волн, включая и световые. Если в качестве Ответчика выступает БПЛА, то нет необходимости менять одновременно ключи у всех БПЛА, так как каждый аппарат настроен на свой инфообмен с Запросчиком *только* по своему простому числу PN_m, порождающему свои КШ.

Для наглядного понимания схемы инфообмена З↔О запишем схему "З" для опроса незнакомого "О":

КШ(PN(1))→[№{A_1(PN(1))}; (N_1,k_1), [(c_nz_n)(x_ny_n),(a_nb_n)])=ВАМ(1,τ(1)).

БПЛА, имеющей регистрационный номер (m) должен ответить по схеме:

КШ(N(1)(k_1↔k_m); PN(1)→PN(m)→[№{A_m(PN(m))}; (N_m,k_m), [(c_nz_n)(x_ny_n),(a_nb_n)]$_m$) = ВАМ(m,τ(m)).

Пояснение к схеме. Запись N(1)(k_1↔k_m); означает, что вначале происходит краткий обмен паролями на языке ВАМ(1). После подтверждения, что это "свой" Ответчик, Запросчик посылает ему инструкции по созданию своей ВАМ(m,τ(m)). Интервальная ВАМ(τ(m)) необязательна, но может использоваться изначально для дублирования сообщения или при необходимости, когда амплитудная схема АМ искажается противником или блокируется.

Поясним эту схему обмена на примере. Пусть Запросчик (1) использует простое число $\mathbf{2}^{0.5}$ (напоминаем, что ТПЧ начинается с двойки) и трансцендентное число $\boldsymbol{E}$ (постоянную Эйлера 2.7182818284590945235360...). Используем самый простейший алгоритм (1) $A_1=2^{0.5}$ и $A_2=\boldsymbol{E}$ и метки k_1=10, k_2=12. Создаем по необходимым формулам ВАМ(1,m) аналогично тому, как создавалась ТАМ. Часть таблицы (основная третья колонка десятичных кодов) для букв русского алфавита ВАМ(1,m) приведена ниже в Приложении 2.

Напоминаем, что две последние целые цифры (x_n,y_n) переводятся в двоичную систему. Дробь после целого числа обеспечивает перевод на новое место локализации символов русского алфавита из второй колонки. Например, десятичное число Таблицы 2 в Приложении 2 "8037,01" равно отсутствию точек впереди, далее идут "тире-точки" от десятичного числа (37_2 = 100101= тире, 2 точки, тире, точка, тире), что соответствует1-ой букве "А".

Соответствие с алфавитом и другими символами, представлены для удобства парой (a_n,b_n), и записаны в виде десятичной дроби. Далее на этом языке ВАМ(1) "З" может опросить незнакомый дрон: "Ты кто?" (используем 2-ой столбец) матрицы Приложения 2. "8849,20; 8237,29; 8708,12; 8849,20; 8817,16; 8212, 54" и послать следующее закодированное сообщение, добавив дополнительно 3 цифры s_{1m},s_{2m},s_{3m}, например, 5,7,9 в соответствии со своим "цветом" 8. Эти три цифры указывают на перемешивание букв (33), цифр (10) и служебных символов (17)) "своему" БПЛА. Если дрон "свой", то он "понимает" язык ВАМ(1) и отвечает "З" уже по другой азбуке ВАМ(m), и использует для кодировки уже число $\boldsymbol{E}$, выбрав в нём для этого транс-числа другую метку k_m=12. Сообщение "Я свой" в десятичной кодировке на языке ВАМ(m) выглядит так (для ответа используется 4-ый столбец матрицы Приложения 2): "8574,33; 8959,19; 8353,03; 8390,16; 8942, 11"

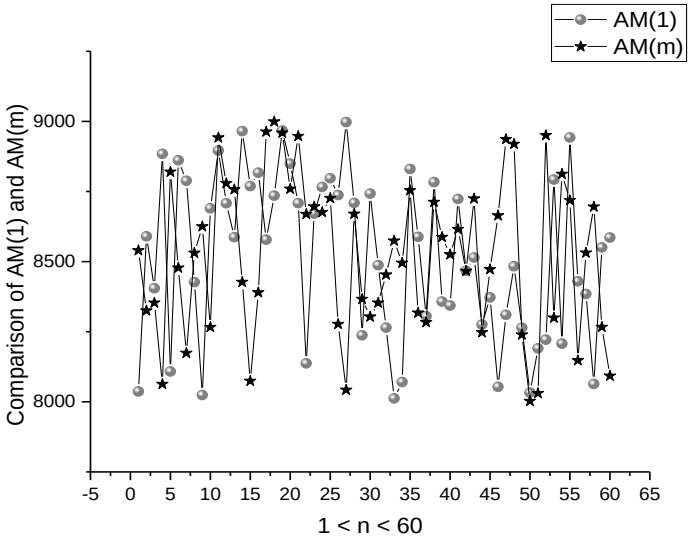

Рис.2. Так выглядят алфавиты ВАМ(1) – серые кружки и ВАМ(m) – черные звездочки для третьей стороны. Эти кодовые десятичные комбинации, превращенные в точки и тире, принимаются на приемники противника уже в виде передаваемой "морзянки" как некий шум

Эти алфавиты можно "спрятать" в исходный ("контейнерный") шум с отношением $S/N<1$ (отдельная задача, решенная ранее автором РРН), незаметно для противника или "зашить" нужную комбинацию ВАМ(m) в электронику опрашиваемого дрона непосредственно на заводском предприятии.

4. Основные выводы

1. На основе анализа ТАМ можно создать оригинальную (и, подчеркнем, принципиально новую) систему кодировки исходных алфавитов по сравнению с общеизвестными [1-3] и найти взаимно однозначное соответствие между точками, тире, двоичными кодами и десятичными наборами цифр. Более того, если теперь двигаться в противоположном направлении от десятичных (закодированных) цифр через двоичные коды к точкам и тире, то можно получить как минимум 10^3 различных ВАМ. Если теперь заменить цифру "8" в начале десятичного кода на произвольную цифру (в интервале [0-9]) - "окрашенная" АМ), то число ВАМ вырастет до величины 10^4. Более того, для обмана третьей стороны можно перед "окрашенным" числом "c_n" вводить "пустые" комбинации из произвольных цифр. Эти пустые комбинации необходимы также для борьбы с помехами [4]. Более того, эффективная передача помехоустойчивых двоичных кодов, основанная на голограммах множества точек рассмотрена в работах [5-6]. Эти методы можно применить к передаче точек-тире созданной ВАМ. Эти ВАМ можно "привязать" к различным существующим алфавитам. В частности, русский алфавит дает 60 = (33+10+17) символов. Английский алфавит дает (25+10+17) =52 символа. Понятно, что китайские иероглифы дадут ещё более непредсказуемые и гигантские комбинации. Число возможных комбинаций (для русских символов) определится выражением (4). Заметим также, что цифра 60 - условная цифра. К ней можно добавить буквы английского алфавита, сленговые и кодовые выражения и пр. Полный список всех символов формируется по соглашению между двумя сторонами.

2. Внимательный анализ существующей азбуки Морзе показывает, что соответствие между цифрами, точками, тире и символами привязываемого алфавита носит *условный* характер. Поэтому учет этой условности приводит к числу (выражение (4)), которое, как следует из этого выражения, можно менять различными комбинаторными способами как минимум 60 раз (для русского алфавита).

3. Следующий этап по повышению криптостойкости предлагаемого кода связан с тем свойством, что можно каждое простое число (они распределены на бесконечном множестве!) с помощью простых алгоритмов типа (5) превратить в ПСП заданной длины. Как уже подчеркивалось выше, нет необходимости передавать сами ПСП. Эти последовательности по возможности заменяются соответствующими алгоритмами, занимающими значительно меньший объём оперативной памяти. Эти алгоритмы также распределены на бесконечном множестве, так как каждому простому числу может соответствовать свой алгоритм. Выбор метки k уже на полученной бесконечной последовательности фиксированной длины N также является *условным* и пара Запрос(k,N_b)↔Ответ(k',N_b') выбирается по предварительному соглашению двумя переговаривающимися сторонами. Эти новые возможности значительно усиливают криптостойкость предлагаемого кода.

4. Как уже упоминалось в тексте, предлагаемый автором способ кодирования можно назвать одним кратким словом, как система "ВАМПИР" (Временная Азбука Морзе – Почти "Идеальное" Решение). Разумеется, этот алгоритм, представленный в виде некой идеальной схемы, необходимо будет скорректировать уже специалистам и конструкторам, разбирающимися детально в электронике, и проверить его уже на макете в "полевых" условиях на российской элементной базе, выбрав из всех предоставляемых возможностей самое приемлемое/оптимальное решение. Поэтому, в этой статье *не* были рассмотрены следующие вопросы, связанные частично уже с практической реализацией предлагаемой системы, в частности: (а) обоснование стойкости протокола и оптимальное хранение всей структуры в оперативной памяти; (б) влияние помех различного рода и передача помехоустойчивой двоичной информации; (в) проверка ПСП на статистических NIST-тестах и др.

5. Можно добавить к вышенаписанному также следующий важный фрагмент. Можно разработать алгоритмы и программы, превращающие *произвольный* (желательно стабильно воспроизводимый для двух сторон) "шум"(а точнее - случайные флуктуации) в стабильно-воспроизводимую АМ! Т.е. при необходимости можно варьировать ТПЧ ↔ "Случайные Флуктуации" (включая также широкий класс детерминированных функций, известных предварительно только для двух сторон) и использовать выбранный калиброванный "шум" как некий "блокнот" для "записи" ВАМ на случайных флуктуациях в соответствии с формулой (3).

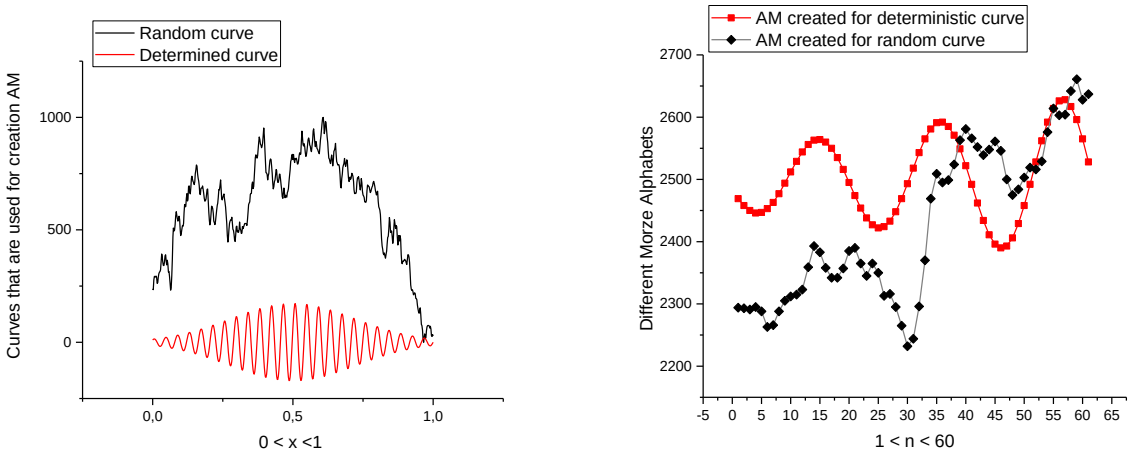

Рис.3. Исходные кривые: случайная кривая и детерминированная кривая (слева) и выделенные из них ВАМ (справа). Соответствие цветов (черный и красный) на рисунках справа и слева одинаковое

Эту идею можно использовать для целей стеганографии. Действительно, любое изображение представляет собой случайные флуктуации. Поэтому, вводя определенную систему координат, можно выделить характерные 60 точек (x_1,x_2,…,x_{60}) и получить соответствующие им случайные точки (y_1,y_2,…,y_{60}). Этот минимальный набор точек {y_n, n=1,2,…,60}, по алгоритмам описанными выше, можно превратить в некую комбинацию (закрытую для третьей стороны) и использовать полученную ВАМ для обмена сообщениями. Например, можно передать скрытно координаты целей, расшифровку карт противника и другую информацию по "стенографической" ВАМ, специально созданную из заданного изображения (это может быть художественная картина, карта местности, денежная купюра и пр.), копии которых имеются у двух сторон, участвующих в информационном обмене.

5. Благодарности

В процессе написания статьи неоценимую помощь автору оказали Туринцев И.В, Матвеев С.В., Борщевский А.И., Кузеев Р.Р. Их советы и замечания автор попытался принять во внимание, что несомненно улучшило качество представленной статьи.

Литература

1. Лось, А.Б. Криптографические методы защиты информации / А.Б. Лось, А.Ю. Нестеренко, М.И. Рожков. - 2-е изд., доп. – Москва: Юрайт, 2016. – 473 с.

2. Алферов, А.П. Основы криптографии: учебное пособие / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин. – 2-е изд., испр. и доп. – Москва: Гелиос АРВ, 2002. – 480с.

3. Панасенко, С. П. Алгоритмы шифрования: специальный справочник // С.П. Панасенко. – СПб.: БХВ-Петербург, 2009. – 576 с., ISBN 978-5-9775-0319-8.

4. Морелос-Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса. – Москва: Техносфера, 2005. – 305 с.

5. Тимофеев, А.Л. Построение помехоустойчивого кода на базе голографического представления произвольной цифровой информации / А.Л. Тимофеев, А.Х. Султанов // Компьютерная оптика. – 2020. – Т. 44, № 6. – С. 978-984. – DOI: 10.18287/2412-6179-CO-739.

6. Тимофеев, А. Л. Использование спектрального подхода при обработке изображений и произвольных данных / А.Л. Тимофеев, А.Х. Султанов, И.К. Мешков, А.Р. Гизатулин // Информационно-управляющие системы. – 2022, № 4. – С.1-11. doi:10.31799/1684-8853-2022-4.

Приложение 1

Таблица 1. Матрица кодов ТАМ (c_n=2)

Буквы русского алфавита (33 символа+10 цифор+17 вспомогательных символов) представлены в виде матрицы, имеющей 60 строк и 6 столбцов):

Русский алфавит

n	Русский Алфавит	Код в десятичном формате	Символы ТАМ $(x_n y_n)_2$	Двоичные коды $(x_n y_n)_2$	Длительности
1	А	2101.01	● –	**01**	τ_{01}
2	Б	2008.02	· – ● ● ●	**1000**	$\tau_{10}(\tau_{00})^2$
3	В	2103.03	● – –	**011**	$\tau_{01}\tau_{11}$
4	Г	2006.04	– – ●	**110**	$\tau_{11}\tau_{10}$
5	Д	2004.05	– ● ●	**100**	$\tau_{10}\tau_{00}$
6	Е	2100.06	●	**0**	τ_0
7	Ё	2100.07	●	**0**	τ_0
8	Ж	2301.08	● ● ● –	0001	$(\tau_{00})^2$
9	З	2012.09	– – ● ●	1100 -	$\tau_{11}\tau_{10}\tau_{00}$
10	И	2200.10	● ●	00	τ_{00}
11	Й	2107.11	● – – –	0111	$\tau_{01}(\tau_{11})^2$
12	К	2005.12	– ● –	101	$\tau_{10}\tau_{01}$
13	Л	2104.13	● – ● ●	0100	$\tau_{01}\tau_{10}\tau_{00}$
14	М	2011.14	– ● – –	1011	$\tau_{10}\tau_{01}\tau_{11}$
15	Н	2002.12	– ●	10	τ_{10}
16	О	2007.16	– – –	111	$(\tau_{11})^2$
17	П	2106.17	● – – ●	0110	$\tau_{01}\tau_{11}\tau_{10}$
18	Р	2102.18	● – ●	010	$\tau_{01}\tau_{10}$
19	С	2300.19	● ● ●	000	$(\tau_{00})^2$
20	Т	2001.2	–	1	τ_1
21	У	2201.21	● ● –	001	$\tau_{00}\tau_{01}$
22	Ф	2202.22	● ● – ●	0010	$\tau_{00}\tau_{01}\tau_{10}$
23	Х	2400.23	● ● ● ●	0000	$(\tau_{00})^3$
24	Ц	2010.24	– ● – ●	1010	$\tau_{10}\tau_{01}\tau_{10}$
25	Ч	2014.25	– – – ●	1110	$(\tau_{11})^2\tau_{10}$
26	Ш	2015.26	– – – –	1111	$(\tau_{11})^3$
27	Щ	2013.27	– – ● –	1101	$\tau_{11}\tau_{10}\tau_{01}$
28	Ъ	2126.28	● – – ● – ●	011010	$\tau_{01}\tau_{11}\tau_{10}\tau_{01}\tau_{10}$
29	Ы	2011.29	– ● – –	1011	$\tau_{10}\tau_{01}\tau_{11}$
30	Ь	2009.3	– ● ● –	1001	$\tau_{10}\tau_{00}\tau_{01}$
31	Э	2204.31	● ● – ● ●	00100	$\tau_{00}\tau_{01}\tau_{10}\tau_{00}$
32	Ю	2203.32	● ● – –	0011	$\tau_{00}\tau_{01}\tau_{11}$
33	Я	2105.33	● – ● –	0101	$\tau_{01}\tau_{10}\tau_{01}$

Цифры (0-9)

n	цифры	Код в 10-ом формате	Символы ТАМ ($x_n y_n$)	Двоичные коды $(x_n y_n)_2$	Длительно-сти
34	0	2031.34	– – – – –	11111	$(\tau_{11})^4$
35	1	2115.35	● – – – –	01111	$\tau_{01}(\tau_{11})^3$
36	2	2207.36	● ● – – –	00111	$\tau_{00}\tau_{01}(\tau_{11})^2$
37	3	2303.37	● ● ● – –	00011	$(\tau_{00})^2\tau_{01}\tau_{11}$
38	4	2401.38	● ● ● ● –	00001	$(\tau_{00})^3\tau_{01}$
39	5	2500.39	● ● ● ● ●	00000	$(\tau_{00})^4$
40	6	2016.4	– ● ● ● ●	10000	$\tau_{10}(\tau_{00})^3$
41	7	2024.41	– – ● ● ●	11000	$\tau_{11}\tau_{10}(\tau_{00})^2$
42	8	2028.42	– – – ● ●	11100	$(\tau_{11})^2\tau_{10}\tau_{00}$
43	9	2030.43	– – – – ●	11110	$(\tau_{11})^3\tau_{10}$

Вспомогательные Символы (1-17)

n	символы	Код в 10-ом формате	Символы ТАМ ($x_n y_n$)	Двоичные коды $(x_n y_n)_2$	Длительности
44	**.**	2600.44	● ● ● ● ● ●	000000	$(\tau_{00})^5$
45	**,**	2121.45	● – ● – ● –	010101	$\tau_{01}\tau_{10}\tau_{01}\tau_{10}\tau_{01}$
46	**:**	2056.46	– – – ● ● ●	111000	$(\tau_{11})^2\tau_{10}(\tau_{00})^2$
47	**;**	2042.47	– ● – ● – ●	101010 -	$\tau_{10}\tau_{01}\tau_{10}\tau_{01}\tau_{10}$
48	**(**	2045.48	– ● – – ● –	101101	$\tau_{10}\tau_{01}\tau_{11}\tau_{10}\tau_{01}$
49	**)**	2045.49	– ● – – ● –	101101	$\tau_{10}\tau_{01}\tau_{11}\tau_{10}\tau_{01}$
50	**'**	2130.5	● – – – – ●	01111	$\tau_{01}(\tau_{11})^3\tau_{10}$
51	**"**	2118.51	● – ● ● – ●	010010	$\tau_{01}\tau_{10}\tau_{00}\tau_{01}\tau_{10}$
52	**–**	2032.52	– ● ● ● ● –	100001	$\tau_{10}(\tau_{00})^3\tau_{01}$
53	**/**	2018.53	– ● ● – ●	10010	$\tau_{10}\tau_{00}\tau_{01}\tau_{10}$
54	**?**	2212.54	● ● – – ● ●	001100	$\tau_{00}\tau_{01}\tau_{11}\tau_{10}\tau_{00}$
55	**!**	2051.55	– – ● ● – –	110011	$\tau_{11}\tau_{10}\tau_{00}\tau_{01}\tau_{11}$
56	@	2126.56	● – – ● – ●	011010	$\tau_{01}\tau_{11}\tau_{10}\tau_{01}\tau_{10}$
57	**err** ошибка	2800.57	● ● ● ● ● ● ● ●	00000000	$(\tau_{00})^7$
58	**endc** конец связи	2205.58	● ● – ● –	00101	$\tau_{00}\tau_{01}\tau_{10}\tau_{01}$
59	bll пробел между буквами	2300.59	● ● ●	000	$(\tau_{00})^2$
60	Blw пробел между словами	2700.60	● ● ● ● ● ● ●	$(0)^7$	$(\tau_{00})^6$

Приложение 2

Матрица кодов ВАМ**,** полученная для чисел $\mathbf{2^{0.5}}$ и **E** в десятичной кодировке (третий столбец матрицы), (c_n=8)

Номер по порядку и буква алфавита	Кодировка для $\mathbf{2^{0.5}}$ ВАМ(1)	Перетасовка дробной части (a_nb_n)	Кодировка для **E** ВАМ(m)	Перетасовка дробной части (a_nb_n)
1-А	8037,01	8037,06	8540,01	8540,03
2-Б	8590,02	8590,05	8325,02	8325,02
3-В	8405,03	8405,04	8353,03	8353,01
4-Г	8884,04	8884,03	8063,04	8063,04
5-Д	8108,05	8108,02	8820,05	8820,05
6-Е	8861,06	8861,01	8478,06	8478,06
7-Ё	8788,07	8788,07	8174,07	8174,11
8-Ж	8427,08	8427,08	8531,08	8531,10
9-З	8024,09	8024,09	8625,09	8625,09
10-И	8690,1	8690,1	8266,1	8266,08
11-Й	8896,11	8896,11	8942,11	8942,07
12-К	8708,12	8708,12	8779,12	8779,33
13-Л	8587,13	8587,13	8757,13	8757,32
14-М	8965,14	8965,14	8427,14	8427,31
15-Н	8769,15	8769,15	8074,15	8074,3
16-О	8817,16	8817,16	8390,16	8390,29
17-П	8578,17	8578,17	8963,17	8963,28
18-Р	8735,18	8735,18	8999,18	8999,27
19-С	8967,19	8967,19	8959,19	8959,26
20-Т	8849,2	8849,2	8759,2	8759,25
21-У	8708,21	8708,21	8947,21	8947,24
22-Ф	8137,22	8137,22	8669,22	8669,23
23-Х	8671,23	8671,23	8696,23	8696,22
24-Ц	8766,24	8766,24	8676,24	8676,21
25-Ч	8797,25	8797,25	8726,25	8726,2
26-Ш	8737,26	8737,26	8277,26	8277,19
27-Щ	8997,27	8997,27	8042,27	8042,18
28-Ъ	8709,28	8709,28	8670,28	8670,17
29-Ы	8237,29	8237,29	8366,29	8366,16
30-Ь	8742,3	8742,3	8303,3	8303,15
31-Э	8487,31	8487,31	8353,31	8353,14
32-Ю	8264,32	8264,32	8453,32	8453,13
33-Я	8012,33	8012,33	8574,33	8574,12

METHOD FOR CREATING AN ADDITIONAL COMMUNICATION CHANNEL, BASED ON TEMPORARY MORSE CODES

R.R. Nigmatullin

Kazan National Research Technical University named after A.N. Tupolev - KAI,
Department of Radio Electronics and Information and Measuring Technology

10, st. K. Marx, Kazan, 420110, Russian Federation

Abstract. This paper discusses new coding methods based on sets of Morse code. These sets of temporal Morse codes (TMC) can be successfully applied, according to the author of the paper in the SW and USW bands. Moreover, this binary system (based on dots and dashes) extends the capabilities of the existing binary system. Infinite pseudorandom sequences derived from prime numbers can be attached to this system. Thus, three sets: (a) an uncountable set of prime numbers, (b) infinite sets of pseudorandom numbers obtained from prime numbers by extraction of product roots (also forming a set of natural numbers), and (c) TMCs attached to them, give in the end a sufficiently reliable system, protected against code breaking, approaching the "ideal" solution. Note also that the proposed system has short crypto-keys convenient for transmission and can be partially used in existing cryptographic systems for protection, in particular, of selected information blocks.

Keywords: Temporal Morse code; Sets of prime numbers; Sets of irrational and transcendental numbers; Compact crypto-keys; Sets of complementary communication channels.

Статья представлена в редакцию 29 декабря 2023г.