МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Радиоуправления и связи»

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

по дисциплине

«Защищенные системы передачи и хранения информации»

Специальность подготовки – 11.04.02 «Инфокоммуникационные технологии и системы связи»

Специальность (профиль) подготовки
Интеллектуальные системы и сети телекоммуникаций

Квалификация выпускника – магистр
Форма обучения – очная

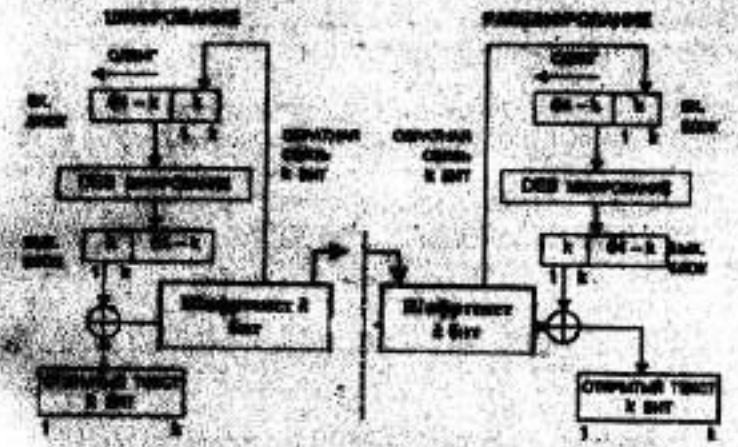
Рязань 2025г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА

ЗАЩИЩЕННЫЕ СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Методические указания
к лабораторным работам



Рязань 2021

УДК 621.396

Защищенные системы передачи информации: методические указания к лабораторным работам /Рязан. гос. радиотехн. ун-т; сост.: С.Н. Кириллов, В.Т. Дмитриев. - Рязань, 2021. – 48 с.

Приведены указания к 4 лабораторным работам, посвященным изучению методов организации защищенного канала передачи информации. Первая лабораторная работа посвящена получению навыков создания зашифрованного сообщения при помощи алгоритма шифрования Эль-Гамаля. Вторая лабораторная работа посвящена изучению методов и алгоритмов помехоустойчивого кодирования. Третья лабораторная работа посвящена изучению основных алгоритмов маскирования речевой информации и их сравнительному анализу. Четвертая лабораторная работа посвящена анализу некоторых режимов работы ГОСТ 28147-89.

Предназначены для студентов специальности 110501 «Радиоэлектронные системы и комплексы», обучающихся по дисциплине «Защищенные системы передачи информации».

Табл. 7. Ил. 19. Библиогр.: 34 назв.

Защита информации, шифры гаммирования, шифры перестановок, шифры замены

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра радиоуправления и связи Рязанского государственного радиотехнического университета (зав. кафедрой проф. С.Н. Кириллов)

Лабораторная работа № 1

АЛГОРИТМ ШИФРОВАНИЯ ЭЛЬ-ГАМАЛЯ

ЦЕЛЬ РАБОТЫ

Получение навыков создания зашифрованного сообщения при помощи алгоритма шифрования Эль-Гамаля.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Алгоритм шифрования Эль-Гамаля является альтернативой алгоритму RSA и при равном значении ключа обеспечивает ту же криптостойкость. Эль-Гамаль усовершенствовал алгоритм Диффи – Хеллмана и получил алгоритмы для шифрования и для обеспечения аутентификации. Алгоритм основан на проблеме поиска дискретного логарифма. Если возводить целое число в степень достаточно легко, то восстановить целочисленный аргумент по значению (то есть найти логарифм) довольно трудно [2].

Для генерации пары ключей сначала выбирается простое число p и два случайных числа, g и x, оба меньше p. Затем вычисляется

$$y = g^x \bmod p.$$

Открытым ключом являются y, g и p. Величины g и p можно сделать общими для группы пользователей. Закрытым ключом является x.

Для шифрования сообщения M сначала выбирается случайное число k, взаимно простое с $(p - 1)$. Затем вычисляются

$$a = g^k \bmod p,$$
$$b = y^k M \bmod p.$$

Пара (a, b) является шифротекстом. Получаемый шифротекст в два раза длиннее открытого текста. Для дешифрирования (a, b) вычисляется

$$M = b/a^x \bmod p.$$

Так как $a^x \equiv g^{kx} \pmod p$ и $b/a^x = y^k M/a^x \equiv g^{xk} M/ g^{kx} = M \pmod p$, то все действительно так и получается. Или иначе:

$a^x \bmod p = y^k \bmod p \Rightarrow (g^k \bmod p)^x \bmod p = (g^x \bmod p)^k \bmod p$.

Каждая подпись или шифрование алгоритмом Эль-Гамаля требует нового значения k, и это значение должно быть выбрано

случайным образом. Если когда-нибудь злоумышленник раскроет k, он сможет раскрыть закрытый ключ x. Если злоумышленник когда-нибудь сможет получить два сообщения, подписанные или зашифрованные с помощью одного и того же k, то он сможет раскрыть x, даже не зная значение k [3].

Рассмотрим алгоритм криптосистемы Эль-Гамаля.

Выбираем открытый ключ p и g:

p – простое число (может быть общим для группы пользователей),

$g < p$ (может быть общим для группы пользователей).

Выбираем закрытый ключ $x < p$.

Вычисляем $y = g^x \bmod p$

Шифрование:

выбираем случайное k, которое взаимно простое с p–1;

a (шифротекст) $= g^k \bmod p$,

b (шифротекст)$= M (y^k \bmod p)$.

Дешифрование:

M (открытый текст) $= b/a^x \bmod p$.

Пусть имеются абоненты A, B, C, ..., которые хотят передавать друг другу зашифрованные сообщения, не имея никаких защищенных каналов связи. Фактически здесь используется схема Диффи-Хеллмана, чтобы сформировать общий секретный ключ для двух абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ. Для каждого следующего сообщения секретный ключ вычисляется заново.

Для всей группы абонентов выбираются некоторое большое простое число p и число g такие, что различные степени g суть различные числа по модулю p. Числа p и g передаются абонентам в открытом виде. Затем каждый абонент группы выбирает свое секретное число c_i, $1 < c_i < p-1$, и вычисляет соответствующее ему открытое число d_i:

$$d_i = g^{c_i} \bmod p. \qquad (1)$$

В результате получаем таблицу:

Ключи пользователей в системе Эль-Гамаля

Абонент	Секретный ключ	Открытый ключ
A	c_A	d_A
B	c_B	d_B
C	c_C	d_C

Покажем теперь, как A передает сообщение m абоненту B. Будем предполагать, что сообщение представлено в виде числа $m < p$.

Шаг 1. A формирует случайное число k, $1 < k < p - 2$, вычисляет числа

$$r = g^k \bmod p, \qquad (2)$$

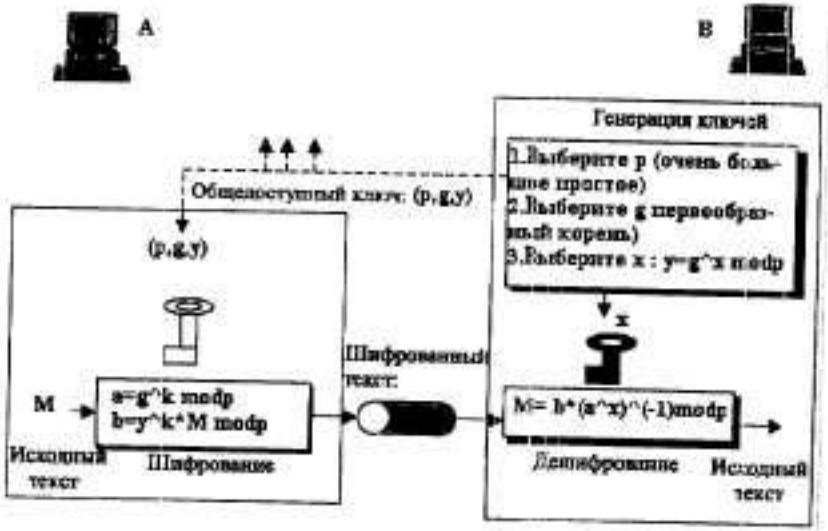
$$e = m \cdot d_B^k \bmod p \qquad (3)$$

и передает пару чисел (r,e) абоненту B.

Шаг 2. B, получив (r,e), вычисляет

$$m' = e \cdot r^{p-1-C_B} \bmod p. \qquad (4)$$

Схема шифрования



Свойства шифра Эль-Гамаля

1. Абонент B получил сообщение, т.е. $m' = m$.

2. Противник, зная p, g, d_B, r и e, не может вычислить m.

Пример. Передадим сообщение $m = 15$ от A к B. Выберем параметры: $p = 23$, $g = 5$. Пусть абонент B выбрал для себя секретное число $c_B = 13$ и вычислил по (1)

$$d_B = 5^{13} \bmod 23 = 21.$$

Абонент A выбирает случайно число k, например k=7, и вычисляет по (2), (3):

$$r=5^7 \bmod 23=17,$$

$$e=15\cdot 21^7 \bmod 23=15\cdot 10 \bmod 23=12.$$

Теперь A посылает к B зашифрованное сообщение в виде пары чисел (17,12). B вычисляет по (4)

$$m'=12\cdot 17^{23-1-13} \bmod 23=12\cdot 17^9 \bmod 23=12\cdot 7 \bmod 23=15.$$

Мы видим, что B смог расшифровать переданное сообщение.

Ясно, что по аналогичной схеме могут передавать сообщения все абоненты в сети. Заметим, что любой абонент, знающий открытый ключ абонента B, может посылать ему сообщения, зашифрованные с помощью открытого ключа d_B, но только абонент B, и никто другой, может расшифровать эти сообщения, используя известный только ему секретный ключ c_B.

Отметим также, что объем шифра в два раза превышает объем сообщения, но требуется только одна передача данных (при условии, что таблица с открытыми ключами заранее известна всем абонентам).

Электронная цифровая подпись схемой Эль-Гамаля

Для того чтобы подписать сообщение M, сначала отправитель хэширует его с помощью хэш-функции $h(\cdot)$ в целое число m:

$m = h(M)$, $1 < m < (P-1)$,

и генерирует случайное целое число K, $1 < K < (P-1)$, такое, что K и $(P-1)$ являются взаимно простыми. Затем отправитель вычисляет целое число a:

$a = G^K \bmod P$

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа X целое число b из уравнения

$m = (X * a + K * b) \pmod{(P-1)}$.

Пара чисел (a,b) образует цифровую подпись S:

$S = (a,b)$,

проставляемую под документом M.

Тройка чисел (M,a,b) передается получателю, в то время как пара чисел (X,K) держится в секрете.

После приема подписанного сообщения (M,a,b) получатель должен проверить, соответствует ли подпись $S = (a,b)$ сообщению M.

Для этого получатель сначала вычисляет по принятому сообщению M число

$m = h(M)$,

т.е. хэширует принятое сообщение M.

Затем получатель вычисляет значение

$A = Y^a a^b \pmod P$

и признает сообщение M подлинным, если и только если

$A = G^m \pmod P$.

Иначе говоря, получатель проверяет справедливость соотношения

$Y^a a^b \pmod P = G^m \pmod P$.

Можно строго математически доказать, что последнее равенство будет выполняться тогда и только тогда, когда подпись $S=(a,b)$ под документом M получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправителем сообщения M был обладатель именно данного секретного ключа X, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ M.

В настоящее время криптосистемы с открытым ключом считаются наиболее перспективными. К ним относится и схема Эль-Гамаля, криптостойкость которой основана на вычислительной сложности проблемы дискретного логарифмирования, где по известным p, g и y требуется вычислить x, удовлетворяющий сравнению:

$$y = g^x \pmod p.$$

ГОСТ Р34.10-1994, принятый в 1994 году в Российской Федерации, регламентировавший процедуры формирования и проверки электронной цифровой подписи, был основан на схеме Эль-Гамаля. С 2001 года используется новый ГОСТ Р 34.10-2001, использующий арифметику эллиптических кривых, определенных над простыми полями Галуа. Существует большое количество алгоритмов, основанных на схеме Эль-Гамаля: это алгоритмы DSA, ECDSA, KCDSA, схема Шнорра.

2. ОПИСАНИЕ РАБОТЫ

- Шифрование

1. Допустим, что нужно зашифровать сообщение M = 5.
2. Произведем генерацию ключей.

Пусть p=11,g=2. Выберем x=8 - случайное целое число x такое, что 1<x<p.

Вычислим $y = g^x \bmod p = 28 \bmod 11 = 3$.

Итак, открытым ключом является тройка (p, g, y) = (11, 2, 3), а закрытым ключом - число x = 8.

3. Выбираем случайное целое число k такое, что 1<k<(p-1). Пусть k=9.
4. Вычисляем число $a = g^k \bmod p = 2^9 \bmod 11 = 512 \bmod 11 = 6$.
5. Вычисляем число $b = y^k M \bmod p = 3^9 5 \bmod 11 = 19683*5 \bmod 11 = 9$.
6. Полученная пара (a,b) = (6,9) является шифротекстом.

- Расшифрование

1. Необходимо получить сообщение M = 5 по известному шифротексту (a, b) = (6, 9) и закрытому ключу x = 8.
2. Вычисляем M по формуле: $M = b(a^x)^{-1} \bmod p = 9(6^8)^{-1} \bmod 11 = 5$.
3. Получили исходное сообщение M = 5.

Так как в схему Эль-Гамаля вводится случайная величина k, то шифр Эль-Гамаля можно назвать шифром многозначной замены. Из-за случайности выбора числа k такую схему еще называют схемой вероятностного шифрования. Вероятностный характер шифрования является преимуществом для схемы Эль-Гамаля, так как у схем вероятностного шифрования наблюдается большая стойкость по сравнению со схемами с определенным процессом шифрования. Недостатком схемы шифрования Эль-Гамаля является удвоение длины зашифрованного текста по сравнению с начальным текстом. Для схемы вероятностного шифрования само сообщение M и ключ не определяют шифротекст однозначно. В схеме Эль-Гамаля необходимо использовать различные значения случайной величины k для шифровки различных сообщений M и M'. Если использовать одинаковые k, то для соответствующих шифротекстов (a, b) и (a', b') выполняется соотношение $b(b')^{-1} = M(M')^{-1}$. Из этого выражения можно легко вычислить M', если известно M.

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

Задача 1
Вычислить открытый ключ y, если p=19, g=5, x=11.

Задача 2
Вычислить шифротекст a и b для сообщения M=5, если p=23, g=7, x=3, k=13.

Задача 3
По известному шифротексту a=10, b=19, используя ключи p=23, g=7, x=5, определить исходное сообщение M.

Задача 4
Произвести шифрование и дешифрование M=7 при заданных ключах p=23, g=5, x=10, k=12.

Задача 5
Сформировать и проверить ЭЦП Эль-Гамаля при следующих начальных условиях: p=11, g=2, секретный ключ x=8.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. На чём основан алгоритм шифрования Эль-Гамаля?
2. Как был получен алгоритм Эль-Гамаля? В чём его отличие от алгоритма RSA?
3. Как осуществляется генерация ключей?
4. Что представляет собой алгоритм шифрования по схеме Эль-Гамаля?
5. Что представляет собой алгоритм дешифрования по схеме Эль-Гамаля?
6. Как осуществляется работа по алгоритму в режиме ЭЦП?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Никифоров С.Н. Методы защиты информации. Шифрование данных: учеб. пособие. – СПб.:Лань, 2019. – 160 с.

2. Голиков А.М. Кодирование и шифрование информации в системах связи. Часть 2. Шифрование: учеб. пособие для специалитета. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2020. – 490 с.

3. Башлы П.Н., Бабаш А.В., Баранова Е.К. Информационная безопасность: учеб.-практ. пособие. – М.: Издательский центр Евразийского открытого института, 2009. – 376 с.

4. Нестеров С.А. Основы информационной безопасности: учеб. пособие. – СПб.:Лань, 2019. – 324 с.

5. Краковский Ю. М. Методы защиты информации. – СПб.: Лань, 2021. – 236 с.

6. Сердюков П.Н., Бельчиков А.В., Дронов А.Е. и др. Защищенные радиосистемы цифровой передачи информации. – М.: АСТ, 2006. – 403 с.

7. Золотарёв В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы: справочник/ под ред. чл.-кор. РАН Ю.Б.Зубарева. – М.: Горячая линия – Телеком, 2004. - 126 с.

8. Кириллов С.Н., Крысяев Д.Е., Дмитриев В.Т. Алгоритм классификации типов помехоустойчивого кодирования // 5-я международная научно-техническая конференция «К.Э. Циолковский - 150 лет со дня рождения. Космонавтика. Радиоэлектроника. Геоинформатика». – Рязань: РГРТУ, 2007. – С. 158–159.

9. Прокис Дж. Цифровая связь: пер. с англ./ под ред. Д.Д. Кловского. – М.: Радио и связь, 2000. – 800 с.

10. Слепов Н. Оптиковолоконные системы дальней связи // Электроника: Наука, Технология, Бизнес, 2005. №6. – С.70-74.

11. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. – М.: Техносфера, 2005. – 319 с.

12. Блейхут Р. Теория и практика кодов, контролирующих ошибки. – М.: Мир, 1986. – 575 с.

13. Панасенко С. Алгоритмы шифрования. – СПб.: БХВ-Петербург, 2009. – 576 с.

Лабораторная работа № 2

ИЗУЧЕНИЕ АЛГОРИТМОВ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ НА ОСНОВЕ КОДОВ ХЭММИНГА

ЦЕЛЬ РАБОТЫ

Изучение методов и алгоритмов помехоустойчивого кодирования, а также принципов построения и функционирования помехоустойчивых кодеков на примере кодов Хэмминга.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1. 1. Функциональная схема системы передачи

Обобщенная функциональная схема системы, использующей помехоустойчивое кодирование, приведена на рис. 1. Источник информации формирует двоичный поток, который преобразуется кодером. Модулятор отображает выходные последовательности кодера в непрерывное множество сигналов. В канале связи добавляются помехи, возникают различные искажения сигналов вследствие ограничения по частоте, воздействия нелинейных факторов, многолучевого распространения сигналов и т.д. Демодулятор формирует оценку переданного символа на основе наблюдения выходного сигнала радиоканала. Декодер производит операции, обратные кодированию, и осуществляет исправление ошибок на основании имеющейся избыточности. Выходной сигнал декодера поступает в приемник информации.

Рис. 1. Обобщенная функциональная схема системы

Помехоустойчивое кодирование связано с введением избыточности символов, что при постоянной скорости источника информации приводит к уменьшению длительности символов и при постоянной мощности передатчика – к уменьшению энергии, приходящейся на один символ. При этом вероятность ошибки увеличивается. Однако за счет исправлений ошибок при декодировании результирующая вероятность ошибки на выходе декодера будет меньше, чем при некодированной передаче.

Важной характеристикой кода является его относительная скорость r = k/n, где k и n - число символов соответственно на входе и выходе кодера (k<n). Величина, пропорциональная 1/r, характеризует избыточность кода и определяет коэффициент расширения полосы частот. Показателем эффективности кодирования является энергетический выигрыш (Q), получаемый при кодировании, который определяется как разность между отношениями сигнал/шум при некодированной $q_{нк}^2$ и кодированной $q_к^2$ передаче, обеспечивающей одинаковое значение вероятности ошибки $p_{ош}$: $Q = q_{нк}^2 - q_к^2$.

1. 2. Классификация помехоустойчивых кодов

В настоящее время разработано множество различных классов помехоустойчивых кодов, отличающихся друг от друга избыточностью, структурой, функциональным назначением, алгоритмами кодирования и декодирования и т.д. [11]. Обобщенная классификация помехоустойчивых кодов представлена на рис. 2 [10].

Рис.2. Классификация помехоустойчивого кодирования

Блоковый код состоит из набора векторов фиксированной длины n_k, называемых кодовыми словами. Элементы кодового слова выбираются из алфавита с q элементами. Если алфавит содержит два элемента 0 и 1, код называется двоичным, а элементы любого кодового слова называются битами. Если элементы кодового слова выбираются из алфавита, имеющего q > 2 элементов, код называется не двоичным. В двоичном блоковом коде длиной n_k можно образовать 2^{n_k} кодовых слов, из которых можно выбрать 2^{k_k} кодовых слов ($k_k < n_k$), чтобы сформировать код, здесь k_k - длина информационной части. Результирующий блоковый код обозначается как (n_k, k_k), а отношение $k_k/n_k = r$ называется кодовой скоростью. Блоковые коды широко используются в радиосистемах передачи информации и других областях. Важной характеристикой кодов, исправляющих ошибки, является расстояние Хэмминга [12].

Одними из самых простых являются блоковые коды Хэмминга, представленные Хэммингом в 1950 г. [6]. К данным кодам относятся линейные блоковые коды с параметрами (n,k) вида ($2^m - 1$, $2^m - m - 1$), где m = n – k - число проверочных символов кода. Коды Хэмминга обладают кодовым расстоянием $d_{min} = 3$ и поэтому способны исправлять только одну или обнаружить две ошибки.

Коды Хэмминга обладают очень слабой корректирующей способностью и отдельно практически не используются. Однако применение данных кодов в составе каскадных схем кодирования позволяет получить очень хорошие результаты. Более подробное описание кодов Хэмминга можно получить в [6]. Коды Боуза – Чоудхури – Хоквингема (БЧХ) [6] представляют собой класс линейных циклических кодов, исправляющих кратные ошибки, и являются обобщением ранее описанных кодов Хэмминга. Коды БЧХ обычно задаются через корни порождающего многочлена g(x) степени n – k.

Среди блоковых кодов наиболее широкое применение нашли недвоичные блоковые коды Рида – Соломона, относящиеся к полиномиальным, компоненты кодовых слов которых равны значениям определенных полиномов. Способность исправления одиночных ошибок, а также пакетов ошибок определенной длины и наличие эффективных алгоритмов декодирования объясняют популярность использования данных кодов.

Для защиты от ошибок в радиоканалах также используется сверточное кодирование, являющееся мощным средством борьбы с одиночными ошибками. Основной характеристикой сверточных кодов является длина кодового ограничения, показывающая, на какое максимальное число выходных символов влияет данный информационный символ. Сложность декодирования сверточных кодов по наиболее выгодному с точки зрения реализации алгоритму Витерби возрастает экспоненциально с увеличением длины кодового ограничения. Наибольший выигрыш сверточные коды обеспечивают при одиночных (случайных) ошибках в канале. В каналах с замираниями необходимо использовать сверточное кодирование совместно с перемежением. Недостатком такого типа кодирования является эффект размножения ошибок, то есть при ограниченном числе ошибок в канале связи возникает неограниченное число ошибок декодирования. Появление такого эффекта связано с характеристиками кода, поэтому на практике стараются не использовать коды, склонные к размножению ошибок.

Составные или производные коды дают компромиссное решение задачи; из них основное значение имеют каскадные коды, коды произведения и турбокоды [6]. Производные коды строят на основе некоторого исходного кода, к которому или добавляют символы, увеличивая расстояние, или сокращают часть информационных символов без изменения расстояния, или выбрасывают некоторые символы.

Разработанные в 1993 году параллельные составные коды (турбокоды) обладают высокой эффективностью в области низких отношений сигнал-шум (ОСШ). Они строятся на основе двоичных сверточных кодов и методов итеративного декодирования, эффективность которых повышается с ростом числа итераций. Недостатком такого алгоритма помехоустойчивого кодирования является большое время декодирования.

Каскадные коды основаны на совместном использовании нескольких составляющих кодов, соединенных последовательно, что позволяет существенно повысить эффективность применения кодирования по сравнению с базовыми некаскадными методами. Такие коды значительно превосходят турбокоды при высоких ОСШ.

Простейшим способом комбинирования кодов является последовательное (повторное) кодирование. Реализация этой идеи для двух кодеров показана на рис. 3 [6].

Рис. 3. Принцип организации повторного кодирования

Возможны различные варианты построения составных кодов. Следует соединить выход первого кодера со входом второго кодера, C_1 называется внешним кодом, а C_2 – внутренним кодом. Код C_1, или C_2, или оба могут быть свёрточными или блоковыми кодами. Если G_1 и G_2 – порождающие матрицы компонентных кодов, то порождающая матрица произведения кодов есть их Кронекеровское произведение.

Произведение кодов может быть интерпретировано как последовательное соединение кодеров через перемежитель. Схематически это изображено на рис. 4 [6].

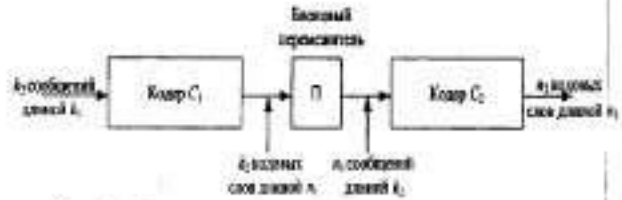


Рис. 4. Структурная схема произведения кодов

Перемежитель является устройством, которое изменяет порядок передачи последовательности символов некоторым взаимно однозначным детерминированным способом. Принцип работы перемежителя показан на рис. 5.

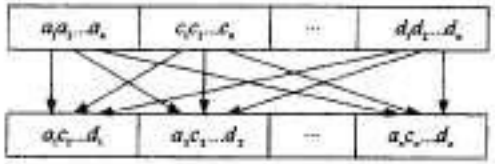


Рис. 5. Принцип работы перемежителя

Перемежители по области применения возможно разделить на следующие группы:

1. Канальный перемежитель.
2. Блоковый и сверточный перемежители.
3. Случайные перемежители.

В большинстве случаев канальные перемежители представляют собой матрицу размером (a,b) и применяются для устранения влияния пакета ошибок в канале связи. Блоковые и сверточные перемежители используются при построении помехоустойчивых алгоритмов, таких как турбокоды, коды произведения и т.п. Случайные перемежители в основном предназначены для обеспечения защиты информации.

Существуют специальные коды, корректирующие пакетные ошибки большей кратности, чем кратность контролируемых случайных ошибок, однако на практике чаще используют перемежение [6]. Различают блоковое, сверточное и псевдослучайное перемежение. При блоковом перемежении биты каждого кодового слова посылаются в канал не друг за другом, а через интервалы, превышающие длину пакета ошибок. В промежутки между битами одного слова вставляются биты других кодовых слов, как показано на рис. 6 [6].

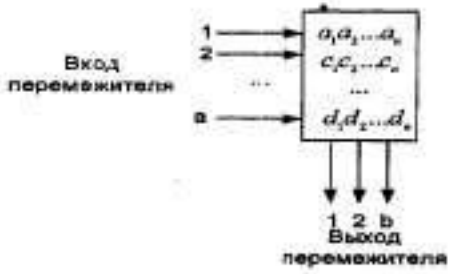


Рис. 6. Блоковый перемежитель

Алгоритм перемежения может быть задан аналитически, путем соответствия выходных бит перемежителя входным битам $c_{вых}(j) = c_{вх}(i)$, определяемым выражением:

$$j = 1 + ((a \cdot i) \bmod b), i = 1,2,3,...,b. \quad (1)$$

Если столбцы на рис. 6 считывать в порядке, определяемом секретным ключом, то получится шифрование данных методом перестановки.

Непрерывное сверточное перемежение по сравнению с блочным способом перемежения данных позволяет более чем в два раза сократить объем памяти. Структурные схемы сверточного перемежителя и деперемежителя показаны на рис. 7.

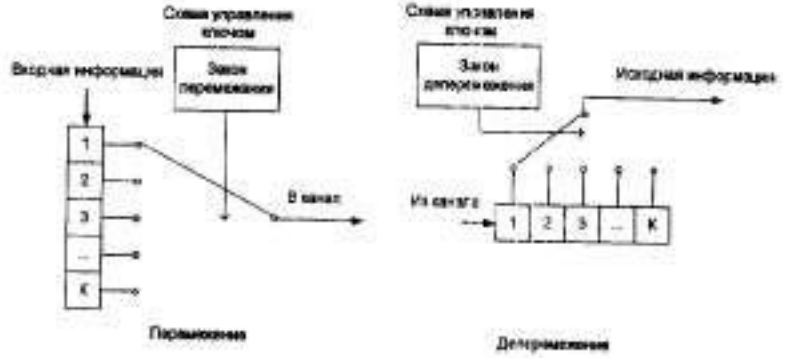


Рис. 7. Сверточный перемежитель и деперемежитель

Глубина перемежения, т.е. перестановка двух соседних бит (байт) пакета после кодирования, составляет K бит (байт). В соответствии с этим параметром кодер перемежения данных состоит из K синхронно коммутируемых по входу и выходу параллельных ветвей.

Декодер перемежения данных строится аналогичным образом, но с обратным порядком включения. Синхронизация и фазирование кодера и декодера перемежения данных производятся по стартовой синхрогруппе пакета - при обнаружении стартовой синхрогруппы входные и выходные коммутаторы ветвей кодера и декодера устанавливаются в положение ветви с номером $N_{ветви} = 0$.

1.3 Код Хэмминга

В коде Хэмминга вводится понятие кодового расстояния d (расстояния между двумя кодами), равного числу разрядов с неодинаковыми значениями. Возможности исправления ошибок связаны с минимальным кодовым расстоянием d_{min}. Исправляются ошибки кратности $r = ent\ (d_{min}-1)/2$, и обнаруживаются ошибки кратности $d_{min}-1$ (здесь ent означает "целая часть"). Так, при контроле

на нечетность $d_{min} = 2$ и обнаруживаются одиночные ошибки. В коде Хэмминга $d_{min} = 3$. Дополнительно к информационным разрядам вводится $L = \log_2 K$ избыточных контролирующих разрядов, где K - число информационных разрядов, L округляется до ближайшего большего целого значения. L-разрядный контролирующий код есть инвертированный результат поразрядного сложения (т.е. сложения по модулю 2) номеров тех информационных разрядов, значения которых равны 1.

Коды Хэмминга позволяют обнаруживать и исправлять все одиночные ошибки (при $d_k = 3$), а также обнаруживать все двойные ошибки, но не исправлять их ($d_k = 4$). Рассмотрим код Хэмминга, обнаруживающий и исправляющий все одиночные ошибки ($q_{обн} - q_{исп} = 1$). В качестве исходного кода берется двоичный код на все сочетания с числом информационных символов k, к которому добавляется m контрольных символов. Таким образом, длина кодовой комбинации $n = k + m$.

При передаче дискретной информации по каналу с помехами может быть либо искажен один из n символов кода, либо кодовая комбинация может быть принята без искажений. Таким образом, при передаче может быть n+1 вариантов, включая передачу без искажений. Используя m контрольных символов, мы должны различить все n+1 случаев. То есть необходимо обеспечить выполнение условия:

$$2^m >= n+1 = k+m+1.$$

В табл. 1 представлена зависимость между k и m, полученная из этого неравенства.

Таблица 1

k	1	2	3	4	5	6	7	8	9	10	11	12	13
m	2	3	3	3	4	4	4	4	4	4	4	5	5

Место расположения контрольных символов значения не имеет. Их можно приписывать перед информационными символами, после них и чередуя информационные символы с контрольными. Для удобства обнаружения искаженных символов целесообразно размещать контрольные символы на местах, кратных степени 2 (то есть на позициях 1, 2, 4, 8 и т.д.). Информационные символы

располагаются на оставшихся местах. Например, для семиэлементной кодовой комбинации можно записать:

$$m_1, m_2, k_4, m_3, k_3, k_2, k_1, \quad (2)$$

где k_4 - старший (четвертый) разряд исходной кодовой комбинации двоичного кода, а k_1 - младший (первый) разряд ДК.

Определенне того, какой символ должен стоять на контрольной позиции (1 или 0), производится по коэффициентам при помощи проверки на четность. Рассмотрим определение коэффициентов на примере комбинаций (2). Составляется табл. 2, в которую выписываются все кодовые комбинации (включая нулевую) для четырехразрядного двоичного кода на все сочетания, а рядом справа, сверху вниз проставляются символы комбинации кода Хэмминга, записанные в последовательности (2). На основе табл. 2 составляется табл. 3, содержащая 3 строки (m = 3). В первую строку табл. 3 записываются символы кода Хэмминга, против которых стоят единицы в младшем (первом) разряде комбинации двоичного кода (k_1) в табл. 2.

Таблица 2

Разряды двоичных чисел				Символы кода
4 (k_4)	3 (k_3)	2 (k_2)	1 (k_1)	
0	0	0	1	m_1
0	0	1	0	m_2
0	0	1	1	k_4
0	1	0	0	m_3
0	1	0	1	k_3
0	1	1	0	k_2
0	1	1	1	k_1

Во вторую строку проверочных коэффициентов записываются символы кода Хэмминга, против которых стоят единицы во втором разряде ДК (k_2) в табл. 2. В третью строку табл. 3 записываются символы, против которых стоят единицы в третьем разряде ДК (k_3 в табл. 2). Число проверок, а значит, число строк табл. 3, равно числу

контрольных символов m = 3. Нахождение состава контрольных символов по коэффициентам табл. 3 при помощи проверки на четность осуществляется следующим образом. Суммируются по модулю 2 информационные символы, входящие в каждую строку табл. 3. Если сумма единиц по данной строке четная, то значение символа m, входящего в эту строку, равно 0, если нечетная, то - 1. При помощи первой строки табл. 3 определяется значение символа m_1, при помощи второй строки - значение символа m_2, третьей - m_3.

Таблица 3

$m_1 = k_4(+)k_3(+)k_1$
$m_2 = k_4(+)k_2(+)k_1$
$m_3 = k_3(+)k_2(+)k_1$

В табл. 3 (+) - оператор сложения по модулю 2.

Пусть нужно передать комбинацию двоичного кода 1101, т.е. k = 4. Согласно табл. 1 число контрольных символов m = 3. Они должны быть размещены на позициях 1, 2 и 4, а информационные - на позициях 3, 5, 6 и 7. Эту последовательность в общем виде можно записать следующим образом:

$$m_1m_2k_4m_3k_3k_2k_1.$$
$$? \; ? \; 1 \; ? \; 1 \; 0 \; 0$$

Для определения значений контрольных символов составим табл. 4 в соответствии с табл. 3.

Таблица 4

$m_1 = 1(+)1(+)1$
$m_2 = 1(+)1(+)1$
$m_3 = 1(+)0(+)1$

Произведём проверку на точность по строкам табл. 4, в результате чего определяются контрольные символы (m_1=1, m_2=0, m_3=0), которые должны дополнять каждую строку до чётного числа единиц. Таким образом, в линию связи будет передана комбинация кода Хэмминга:

$$m_1m_2k_4m_3k_3k_2k_1.$$
$$1 \; 0 \; 1 \; 0 \; 1 \; 0 \; 0$$

Для повышения помехоустойчивости первичного кода необходимо посылать дополнительные контрольные символы, которые увеличивают длину кодовой комбинации, вследствие чего появляются дополнительные или избыточные кодовые комбинации, не используемые непосредственно для передачи информации. Так, семиразрядный код принципиально обеспечивает передачу $N_k = 2^7$=128 кодовых комбинаций. Количество информации в семиразрядном коде Хэмминга k=4, т.е. число полезных информационных комбинаций составляет $N_k = 2^6$=64, остальные 112 кодовых комбинаций из 128 предназначены для обеспечения помехоустойчивости кода и являются запрещёнными. Следовательно, избыточность кода Хэмминга достаточно велика и может быть определена из следующего выражения:

$$I = (n-k)/k = m/k,$$

где I - избыточность кода, m - число контрольных символов, k - число информационных символов, n - общее число символов. Для рассмотренного выше семиразрядного кода Хэмминга I=3/4, причём с увеличением разрядности кода избыточность уменьшается.

2. ОПИСАНИЕ РАБОТЫ

Допустим, что при прохождении сигнала от передатчика к приемнику на него действует аддитивная помеха, эквивалентная добавлению вектора ошибки e. В этом случае при перемножении на проверочную матрицу мы получаем синдром s ошибки [7]:

$$\hat{c}H^T = (c+e)H^T = cH^T + eH^T = eH^T = s.$$

Если число ненулевых разрядов вектора ошибок e не более одного (т.е. вектор ошибок удовлетворяет условию исправления ошибок), то каждому вектору ошибок соответствует свой синдром. Для исправления ошибок используют таблицу синдромов, рассчитанную заранее.

Таким образом, алгоритм передачи сигнала состоит из следующих пунктов.

1. Для всех возможных векторов ошибок рассчитываются их синдромы и составляется таблица синдромов.

2. Производится перемножение информационного слова на порождающую матрицу кода.

3. На приемной стороне принятую последовательность перемножают на проверочную матрицу и получают синдром ошибки.

4. По таблице синдромов находят вектор ошибки.

5. Вычитают из принятого слова вектор ошибки и получают посланную комбинацию.

6. По посланной комбинации восстанавливают переданное информационное сообщение.

3. ПРАКТИЧЕСКАЯ ЧАСТЬ

Для изучения исправляющих свойств кода Хэмминга предлагается использовать код (7,4) с порождающей матрицей

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

и проверочной матрицей

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

1. Необходимо составить программу вычисления кодового слова по информационному слову и порождающей матрице G. Значение информационного слова берется из табл. 5 согласно варианту.

Таблица 5

Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5
0110	0111	1110	1000	1111
1101	0001	0010	0111	0011
1110	1100	0101	1001	1010
0010	1010	1010	1100	1011
Вариант 6	Вариант 7	Вариант 8	Вариант 9	Вариант 10
1110	1100	0001	1010	1010
0110	0001	0110	0001	0011
0001	1110	1100	0000	0010
1110	0101	1110	0111	1111

2. Составьте программу вычисления таблицы синдромов ошибок для заданной проверочной матрицы H.

3. В табл. 6 приведены значения принятых посылок. Вычислите синдром данных посылок, определите наличие ошибки в принятой посылке и, пользуясь таблицей синдромов, разряд, в котором произошла данная ошибка.

Таблица 6

Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5
0001011	1101110	1011011	0110100	0101001
1011100	0111010	1111100	1011101	1000001
0101000	0110010	0010100	1111100	0101101
1001010	1000001	1001111	0110100	1001100
Вариант 6	Вариант 7	Вариант 8	Вариант 9	Вариант 10
1011000	0110001	0011110	0100001	1011001
0100110	1110010	1101001	0111110	1110010
0010100	1011101	0111010	1101001	0101001
1101000	0101000	0001101	1100111	0010100

4. По результатам работы сделайте выводы и отразите их в отчете.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Изобразите структурную схему системы передачи и перечислите функции отдельных ее блоков.

2. Назначение помехоустойчивого кодирования.

3. Перечислите основные характеристики кода.

4. Какой код называется блоковым? Перечислите виды блоковых кодов.

5. Что собой представляют сверточные коды?

6. Что собой представляют каскадные коды?

7. Что собой представляет составной код?

8. Приведите структурную схему и объясните работу сверточного перемежения.

9. Как формируется код Хэмминга? Его основные характеристики.

УКАЗАНИЯ К СОСТАВЛЕНИЮ ОТЧЕТА

Отчет должен содержать:

1) структурную схему системы передачи;
2) составленную программу и результаты расчета синдрома;
3) краткий анализ проведенных исследований и выводы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Березкин Е. Ф. Основы теории информации и кодирования: учеб. пособие. – СПб.:Лань, 2019. – 320 с.

2. Игнатьев Е. Б. Основы криптографии: учеб. пособие. – Иваново: Ивановский государственный энергетический университет имени В.И. Ленина, 2020. – 88 с.

3. Бурсиан Е. Ю., Бураков Д. П. Основы кодирования. Ч. 1: учеб. пособие. – СПб.:Лань, 2018. – 70 с.

4. Лидовский В. В. Основы теории информации и криптографии. – М.: ИНТУИТ, 2016. – 141 с.

5. Басалова Г. В. Основы криптографии: учеб. пособие – М.: ИНТУИТ, 2016. – 282 с.

6. Сердюков П.Н., Бельчиков А.В., Дронов А.Е. и др. Защищенные радиосистемы цифровой передачи информации. – М.: АСТ, 2006. – 403 с.

7. Золотарёв В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы: справочник/ под ред. чл.-кор. РАН Ю.Б.Зубарева. – М.: Горячая линия – Телеком, 2004. – 126 с.

8. Кириллов С.Н., Крысяев Д.Е., Дмитриев В.Т. Алгоритм классификации типов помехоустойчивого кодирования // 5-я международная научно-техническая конференция «К.Э. Циолковский - 150 лет со дня рождения. Космонавтика. Радиоэлектроника. Геоинформатика». – Рязань: РГРТУ, 2007. – С. 158–159.

9. Прокис Дж. Цифровая связь: пер. с англ./ под ред. Д.Д. Кловского. – М.: Радио и связь, 2000. – 800 с.

10. Слепов Н. Оптоволоконные системы дальней связи // Электроника: Наука, Технология, Бизнес, 2005. №6. – С.70-74.

11. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. – М.: Техносфера, 2005. – 319 с.

12. Блейхут Р. Теория и практика кодов, контролирующих ошибки. – М.: Мир, 1986. – 575 с.

Лабораторная работа № 3

АЛГОРИТМЫ МАСКИРОВАНИЯ РЕЧЕВОЙ ИНФОРМАЦИИ В МНОГОКАНАЛЬНЫХ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

ЦЕЛЬ РАБОТЫ

Изучение основных алгоритмов маскирования речевой информации и их сравнительный анализ.

I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В речевых системах связи известны два основных метода закрытия речевых сигналов, разделяющихся по способу передачи по каналам связи: аналоговое скремблирование и дискретизация речи с последующим шифрованием (цифровое скремблирование) Под скремблированием (маскированием) понимают изменение характеристик речевого сигнала таким образом, чтобы полученный сигнал становился неразборчивым и неузнаваемым, занимая ту же полосу спектра, что и исходный. При использовании скремблера обеспечивается защита телефонных переговоров от любых средств съема информации [1].

Наибольшая часть аппаратуры засекречивания речевых сигналов использует в настоящее время метод аналогового скремблирования, поскольку:

- это дешево;
- необходимая для этого аппаратура применяется в большинстве случаев в стандартных телефонных каналах с полосой 3,1 кГц;
- обеспечивается коммерческое качество дешифрованной речи;
- гарантируется достаточно высокая стойкость закрытия.

Аналоговые скремблеры преобразуют исходный речевой сигнал посредством изменения его амплитудных, частотных и временных параметров в различных комбинациях. В аппаратах такого типа используется один или несколько способов аналогового скремблирования из числа следующих:

- скремблирование в частотной области — частотная инверсия (преобразование спектра сигнала с помощью гетеродина и фильтра), частотная инверсия и смещение (частотная инверсия с меняющимся скачкообразно смещением несущей частоты), разделение полосы частот речевого сигнала на ряд поддиапазонов с последующей их перестановкой и инверсией;

- скремблирование по временной области — разбиение блоков или частей речи на сегменты с перемешиванием их во времени с последующим их прямым и (или) реверсивным считыванием;

- комбинация временного и частотного скремблирования [6].

Как правило, все перестановки каким-либо образом выделенных сегментов или участков речи во временной и (или) в частотной областях осуществляются по закону псевдослучайной последовательности, вырабатываемой шифратором по ключу, меняющемуся от одного сообщения к другому.

На стороне приемника выполняется дешифрование цифровых кодов, полученных из канала связи, и преобразование в аналоговую форму. Системы, работа которых основана на таком методе, являются достаточно сложными, поскольку для обеспечения высокого качества передаваемой речи требуется высокая частота дискретизации входного аналогового сигнала и соответственно высокая скорость передачи данных по каналу связи. Каналы связи, которые обеспечивают скорость передачи данных только 2400 Бод, называются узкополосными, в то время как другие, обеспечивающие скорость передачи свыше 2400 Бод, относят к широкополосным.

Аналоговые скремблеры по режиму работы можно разделить на два следующих класса:

- статические, схема кодирования которых остается неизменной в течение всей передачи речевого сообщения;

- динамические, постоянно генерирующие кодовые подстановки в ходе передачи (код может быть изменен в процессе передачи несколько раз в течение каждой секунды).

Очевидно, что динамические скремблеры обеспечивают более высокую степень защиты, поскольку резко ограничивают возможность легкого прослушивания переговоров посторонними лицами.

Преобразование речевого сигнала возможно по трем параметрам: амплитуде, частоте и времени. Считается, что использовать амплитуду нецелесообразно, так как изменяющиеся во

времени затухание канала и отношение сигнал/шум делают сложным точное восстановление амплитуды переданного сигнала. Поэтому практическое применение получили только частотное и временное скремблирование и их комбинации.

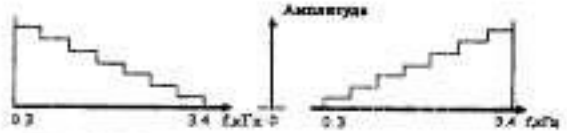


Рис. 1. Принцип работы инвертора спектра речи

Существуют два основных вида частотных скремблеров: инверсный и полосовой. Оба основаны на преобразованиях спектра исходного речевого сигнала для скрытия передаваемой информации и восстановления полученного речевого сообщения путем обратных преобразований. Инверсный скремблер осуществляет преобразование речевого спектра, равносильное повороту частотной полосы речевого сигнала вокруг некоторой средней точки (рис. 1). Однако данный способ обеспечивает невысокий уровень закрытия, так как при перехвате легко устанавливается значение частоты, соответствующее средней точке инверсии в полосе речевого сигнала. Речевой спектр можно также разделить на несколько частотных полос и произвести их перемешивание и инверсию по некоторому правилу (ключу системы). Так функционирует полосовой скремблер (рис. 2).

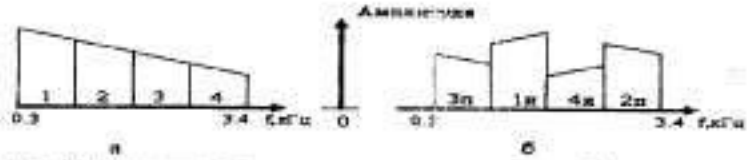


Рис. 2. Принцип работы четырехполосового скремблера речи:
а - исходный речевой спектр; б - преобразованный спектр

Изменение ключа системы позволяет повысить степень закрытия, но требует введения синхронизации на приемной стороне системы. Основная часть энергии речевого сигнала сосредоточена в

небольшой области низкочастотного спектра, поэтому выбор вариантов перемешивания ограничен.

Существенное повышение степени закрытия речи может быть достигнуто путем реализации в полосовом скремблере быстрого преобразования Фурье (БПФ). При этом число допустимых перемешиваний частотных полос значительно увеличивается, что обеспечивает высокую степень закрытия без ухудшения качества речи. Можно дополнительно повысить степень закрытия задержкой различных частотных компонент сигнала на различное время. Пример реализации такой системы показан на рис. 3.

Главным недостатком использования БПФ является возникновение в системе большой задержки сигнала (до 300 мс), обусловленной необходимостью использования весовых функций. Это приводит к затруднениям в работе дуплексных систем связи.

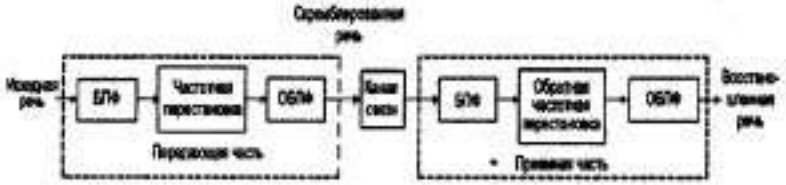


Рис. 3. Основная форма реализации аналогового скремблера речи на основе БПФ

В скремблерах с временной инверсией речевой сигнал делится на последовательность временных сегментов и каждый из них передается инверсно во времени (с конца). Такие скремблеры обеспечивают ограниченный уровень закрытия, зависящий от длительности сегментов. Для достижения неразборчивости медленной речи необходимо, чтобы длина сегмента составляла около 250 мс. Это означает, что задержка системы будет равна примерно 500 мс, что может оказаться неприемлемым в некоторых случаях.

Для повышения уровня закрытия прибегают к способу перестановки временных отрезков речевого сигнала в пределах фиксированного кадра (рис. 4). Правило перестановок является ключом системы, изменением которого можно существенно повысить степень закрытия речи. Остаточная разборчивость зависит от

длительностей отрезков сигнала и кадра и с увеличением последнего уменьшается [7].

Главным недостатком скремблера с фиксированным кадром является большое время задержки системы, равное удвоенной длительности кадра. Этот недостаток устраняется в скремблере с перестановкой временных отрезков речевого сигнала со скользящим окном. В нем число комбинаций возможных перестановок ограничено таким образом, что задержка любого отрезка не превосходит установленного максимального значения. Каждый отрезок исходного речевого сигнала как бы имеет временное окно, внутри которого он может занимать произвольное место при скремблировании. Это окно скользит во времени по мере поступления в него каждого нового отрезка сигнала. Задержка при этом снижается до длительности окна.

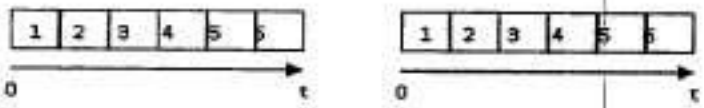


Рис. 4. Схема работы временного скремблера с перестановками в фиксированном кадре

Комбинированный скремблер намного сложнее обычного и требует компромиссного решения по выбору уровня закрытия, остаточной разборчивости, времени задержки, сложности системы и степени искажений в восстановленном сигнале. В качестве примера такой системы рассмотрим скремблер, схема которого представлена на рис. 5, где операция частотно-временных перестановок дискретизированных отрезков речевого сигнала осуществляется с помощью четырех процессоров цифровой обработки сигналов, один из которых может реализовывать функцию генератора случайной последовательности (ключа системы закрытия) [7].

В таком скремблере спектр оцифрованного аналого-цифровым преобразователем (АЦП) речевого сигнала разбивается посредством использования алгоритмов цифровой обработки сигналов на частотно-временные элементы, которые затем перемешиваются на частотно-временной плоскости в соответствии с одним из криптографических

алгоритмов (рис. 6) и суммируются, не выходя за пределы частотного диапазона исходного сигнала.

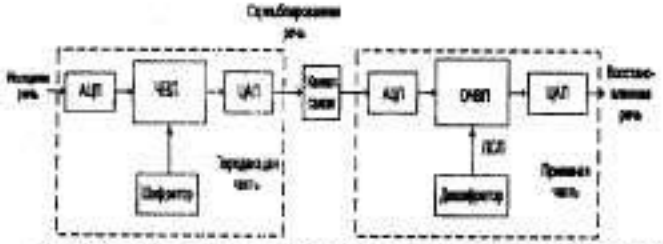


Рис. 5. Структурная схема комбинированного скремблера

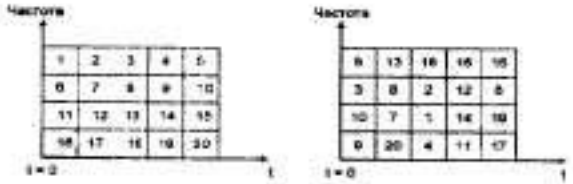


Рис. 6. Принцип работы комбинированного скремблера со стойкостью систем цифрового закрытия речи

На приемном конце производятся обратные операции по восстановлению полученного закрытого речевого сообщения.

Скремблеры всех типов, за исключением простейшего (с частотной инверсией), вносят искажения в восстановленный речевой сигнал. Границы временных сегментов нарушают целостность сигнала, что неизбежно приводит к появлению внеполосных составляющих. Результатом искажения является увеличение минимально допустимого отношения сигнал/шум, при котором может осуществляться надежная связь.

Дискретизация речи с последующим шифрованием

Альтернативным аналоговому скремблированию методом передачи речи в закрытом виде является шифрование речевых сигналов, преобразованных в цифровую форму, перед их передачей. Этот метод обеспечивает более высокий уровень закрытия по

сравнению с описанными выше аналоговыми методами. В основе устройств, работающих по такому принципу, лежит представление речевого сигнала в виде цифровой последовательности, закрываемой по одному из криптографических алгоритмов. Передача данных, представляющих дискретизированные отсчеты речевого сигнала и его параметры, по телефонным сетям, как и в случае устройств шифрования алфавитно-цифровой и графической информации, осуществляется через устройства, называемые модемами. Основной целью при разработке устройств цифрового закрытия речи является сохранение тех ее характеристик, которые наиболее важны для восприятия слушателем [1].

Сохранение формы сигнала требует высокой скорости передачи и соответственно использования широкополосных каналов связи. Для узкополосных каналов требуются устройства, исключающие избыточность речи до ее передачи. Снижение информационной избыточности речи достигается параметризацией речевого сигнала, при которой характеристики речи, существенные для восприятия, сохраняются.

Основной особенностью использования систем цифрового закрытия речевых сигналов является необходимость применения модемов. В принципе возможны следующие подходы при проектировании систем цифрового закрытия речевых сигналов:

- цифровая последовательность параметров речи с выхода вокодерного устройства подается на вход шифратора, где подвергается преобразованию по одному из криптографических алгоритмов, затем поступает через модем в канал связи, на приемной стороне которого осуществляются обратные операции по восстановлению речевого сигнала, в которых задействованы модем и дешифратор. Шифрующие (дешифрующие) функции обеспечиваются либо в отдельных устройствах, либо в программно-аппаратной реализации самого вокодера;

- шифрующие (дешифрующие) функции обеспечиваются самим модемом (так называемый засекречивающий модем) обычно по известным криптографическим алгоритмам типа DES и др. Цифровой поток, несущий информацию о параметрах речи, с выхода вокодера непосредственно поступает на такой модем. Организация связи по каналу аналогична вышеприведенной.

Асинхронное маскирование речи на основе алгоритма Хургина – Яковлева

Существующие методы маскирования РС значительно уменьшают динамический диапазон РС. В силу этого восстановленный на приемной стороне РС обладает достаточно низкой разборчивостью.

Один из путей решения перечисленных выше проблем возможен на основе алгоритма Хургина - Яковлева [11], использующего представление исходной информации в виде отсчетов сигнала и его производной. Использование алгоритма Хургина - Яковлева в интересах маскирования речевых сигналов позволит повысить помехоустойчивость и реализационные возможности системы маскирования.

Алгоритм Хургина - Яковлева рассматривает представление РС с финитным спектром и верхней граничной частотой F в виде совокупности отсчетов сигнала и его N-1 первых производных, взятых с частотой дискретизации $f_д = 2F/N = f_{д.0}/N$, где $f_{д.0} = 2F$ - частота дискретизации, определяемая в соответствии с теоремой В.А. Котельникова. Исходный сигнал f(t) на основе алгоритма Хургина - Яковлева может быть представлен в виде [Ц, 12]:

$$f(t) = \sum_{k=0}^{N-1} \sum_{n=-\infty}^{\infty} f^{(k)}(nN\Delta)(t - nN\Delta)^k \cdot [\mathrm{sinc}(\alpha)]^n / k!,$$

где $\mathrm{sinc}(\alpha)=\sin(\alpha)/\alpha$, $\alpha=\pi(t-nN\Delta)/N\Delta$, $f^{(k)}(nN\Delta)$ – отсчеты k-й производной сигнала, $\Delta = 1/2F$. В [12] рассмотрена возможность использования алгоритма Хургина – Яковлева при N=2, что подразумевает представление исходного РС f(n) в виде прореженных отсчетов сигнала $f_c(n)$ и производной $f_п(n)$. В этом случае устройство (рис. 7) восстановления исходного сигнала включает два канала, в которых осуществляется обработка последовательности отсчетов сигнала $f_c(n)$ и производной $f_п(n)$ в синтезирующих фильтрах с импульсными характеристиками $\mathrm{sinc}^2(\alpha)$ и $(t-nN\Delta)\cdot\mathrm{sinc}^2(\alpha)$ соответственно.

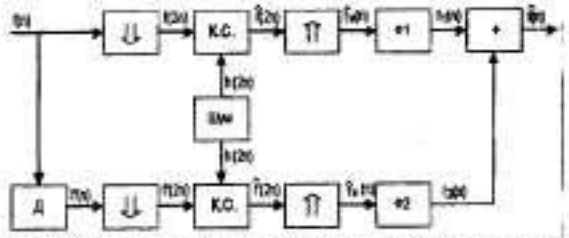


Рис. 7. Схема алгоритма Хургина - Яковлева для N=2

На рис. 7 введены обозначения: Д – дифференциатор, [↓] - дециматор, К.С. – канал связи, [↑] - интерполятор, Ф1 и Ф2 – синтезирующие фильтры сигнала и его производной в алгоритме Хургина - Яковлева, h(2n) - прореженные отсчеты шума в канале, $\hat{f}(2n), \hat{f}'(2n)$ и $\hat{f}_в(n), \hat{f}'_в(n)$ - отсчеты сигнала и производной на входе и на выходе интерполятора соответственно, f(2n), $\hat{f}(2n)$ и f(2n), $\hat{f}(2n)$ - отсчеты сигнала и производной на входе и на выходе канала связи соответственно, $f_1(n)$, $f_2(n)$ - отсчеты сигнала на выходе первого и второго синтезирующих фильтров.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Исследование скремблирования в частотной области

1.1. Запустите программу моделирования.

1.2. Выберите частотные методы и частотный инвертор.

1.3. Откройте звуковой файл "file1.wav", представляющий собой образец человеческой речи.

1.4. Установите высокое отношение сигнал-шум (порядка 40-100 дБ) и N, равное 10.

1.5. Проведите кодирования файла при разных значениях средней частоты инверсии (возвращение к исходному файлу осуществляется нажатием кнопки "Вернуться к исх. файлу").

1.6. При этом после каждого сеанса кодирования прослушайте полученный файл и оцените его остаточную разборчивость путем экспертной оценки по пятибалльной шкале (субъективно).

Зафиксируйте данную оценку в отчет. Кроме того, для каждого кодированного файла фиксируйте в отчет объективные показатели отличия исходного от кодированного файла:

- проследить за изменением ВКФ в течение эксперимента;
- среднее (поотсчетное) различие рабочих сигналов.

1.7. Сделайте вывод, какие настройки скремблера позволяют снизать остаточную разборчивость и повысить уровень закрытия информации для данного типа сигналов.

Субъективную оценку целесообразно учитывать, когда основная энергия преобразованного сигнала лежит в диапазоне частот 40 – 14000 Гц, так как более высокие частоты с трудом различаются человеческим слухом и могут претерпевать значительные искажения в каналах связи.

1.8. Повторите пп. 1.5-1.7 для N, равного 100, 1000, 10000.

1.9. Откройте звуковой файл "file2.wav", представляющий собой образец человеческой речи.

1.10. Повторите для данного файла пп. 1.5-1.8.

1.11. Откройте звуковой файл "file3.wav".

1.12. Повторите для данного файла пп. 1.5-1.8.

1.13. Выберите частотные методы и полосовой скремблер.

1.14. Установите значения M и N порядка 10 и проведите кодирование файла.

1.15. Выполните пп. 1.6, 1.7.

1.16. Путем копирования и вставки из полей для ключевых последовательностей кодека измените (2-7) значений, имитируя работу по подбору ключа.

1.17. Зафиксируйте в отчет условия и результаты данного эксперимента.

1.18. Установите значения M и N порядка 100 и проведите кодирование файла.

1.19. Выполните пп. 1.6, 1.7 и 1.16-1.18.

1.20. Установите значения M и N порядка 1000 и проведите кодирование файла.

1.21. Выполните пп. 1.6, 1.7 и 1.16-1.18.

1.22. Выберите частотные методы и полосовой инвертирующий скремблер.

1.23. Выполните пп. 1.14-1.21.

2. Исследование скремблирования во временной области

2.1. Выберите временные методы и временной инвертор.

2.2. Установите N, равное 10, и откройте звуковой файл "file4.wav".

2.3. Проведите кодирования файла при разных значениях N (100, 1000, 10000).

2.4. Выполните п. 2.3 и пп. 1.6, 1.7.

2.5. Выберите временные методы и временные перестановки.

2.6. Выполните пп. 1.14-1.21.

3. Исследование возможностей скрытной передачи сообщений

3.1. По результатам выполненной работы выберите лучший скремблер и его оптимальные параметры по совокупности описанных выше критериев.

3.2. Откройте звуковой файл "file1.wav".

3.3. Выберите временные методы и режим работы без кодирования.

3.4. Уменьшайте отношение сигнал-шум (и нажимайте кнопку "Кодировать файл") до тех пор, пока еще можно разобрать информацию в сообщении (пока не будет потеряна разборчивость).

3.5. Настройте скремблер в соответствии с п. 3.1 и проведите кодирование. При этом не используйте случаи, когда основная энергия преобразованного сигнала лежит в диапазоне частот 40 – 14 000 Гц.

3.6. Оцените остаточную разборчивость на основе п. 1.6.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение маскиратора.
2. Виды скремблеров, их основные особенности.
3. Искажения сигналов при использовании скремблеров.
4. Достоинства и недостатки скремблеров.
5. Укажите связь между методами закрытия речевых сигналов и степенью секретности.
6. Приведите функциональные схемы аналоговых скремблеров.
7. Достоинства алгоритма Хургина – Яковлева.
8. Схема алгоритма Хургина - Яковлева для N=2.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кириллов С. Н., Дмитриев В. Т. Защита информации в МТКС: учеб. пособие. – Рязань: Рязанский государственный радиотехнический университет, 2018. – 48 с.

2. Руденков Н.А., Пролетарский А.В., Смирнова Е.В., Суровов А.М. Технологии защиты информации в компьютерных сетях. – М.: ИНТУИТ, 2016. – 368 с.

3. Скрипник Д. А. Общие вопросы технической защиты информации. – М.: ИНТУИТ, 2016. – 424 с.

4. Донгак Ш. М. Криптография. Часть II: Практикум. – М.: МИРЭА, 2020. – 64 с.

5. Ермакова А. Ю. Методы и средства защиты компьютерной информации: учеб. пособие. – М.: МИРЭА, 2020. – 223 с.

6. Голиков А. М. Защита информации от утечки по техническим каналам: учеб. пособие. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2015. – 256 с.

7. Тумбинская М.В., Петровский М.В. Комплексное обеспечение информационной безопасности на предприятии: учебник. – СПб.: Лань, 2019. – 344 с.

8. Кириллов С.Н., Малинин Д.Ю. Теоретические основы асинхронного маскирования речевых сигналов: учеб. пособие. – Рязань: РГРТА, 2000. – 80 с.

9. Джейнс Э.Т. О логическом обосновании методов максимальной энтропии// ТИИЭР. 1982. Т.70. №9.

10. Burg J.P. Maximum Entropy Spectral Analysis. – Oklahoma City. OK. 1967.

11. Хургин Я.И., Яковлев В.П. Финитные функции в физике и технике. – М.: Наука, 1971. – 408 с.

12. Кириллов С.Н., Дмитриев В.Т. Реализационные возможности и помехоустойчивость процедуры восстановления сигналов на основе алгоритма Хургина – Яковлева// Радиотехника, 2003. №1. С. 73-75.

Лабораторная работа № 4

РЕЖИМЫ РАБОТЫ ГОСТ 28147-89

ЦЕЛЬ РАБОТЫ

Анализ некоторых режимов работы ГОСТ 28147-89.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

ГОСТ 28147-89 — советский и российский стандарт симметричного шифрования, введённый в 1990 году, также является стандартом СНГ. Полное название— «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».

ГОСТ 28147-89 — блочный шифр с 256-битным ключом и 32 циклами преобразования, оперирующий 64-битными блоками.

Является примером DES-подобных криптосистем, созданных по классической итерационной схеме Фейстеля.

Стандарт отменён на территории России и СНГ с 31 мая 2019 года в связи с принятием новых полностью его заменяющих межгосударственных стандартов ГОСТ 34.12-2018 (описывает шифры «Магма» и «Кузнечик») и ГОСТ 34.13-2018 (описывает режимы работы блочных шифров).

Логика построения шифра и структура ключевой информации ГОСТа

Если внимательно изучить оригинал ГОСТ 28147–89, можно заметить, что в нем содержится описание алгоритмов нескольких уровней. На самом верхнем находятся практические алгоритмы, предназначенные для шифрования массивов данных и выработки для них имитовставки. Все они опираются на три алгоритма низшего уровня, называемые в тексте ГОСТа циклами. Они имеют следующие названия и обозначения:

цикл зашифрования (32-З);
цикл расшифрования (32-Р);
цикл выработки имитовставки (16-З).

В свою очередь, каждый из базовых циклов представляет собой многократное повторение одной единственной процедуры, называемой для определенности далее в настоящей работе основным шагом криптопреобразования.

Таким образом, чтобы разобраться в ГОСТе, надо понять три следующие вещи:

а) что такое основной шаг криптопреобразования;

б) как из основных шагов складываются базовые циклы;

в) как из трех базовых циклов складываются все практические алгоритмы ГОСТа.

Прежде чем перейти к изучению этих вопросов, следует поговорить о ключевой информации, используемой алгоритмами ГОСТа (рис.1). В соответствии с принципом Кирхгофа, которому удовлетворяют все современные известные широкой общественности шифры, именно ее секретность обеспечивает секретность зашифрованного сообщения. В ГОСТе ключевая информация состоит из двух структур данных. Помимо собственно ключа, необходимого для всех шифров, она содержит еще и таблицу замен. Ниже приведены основные характеристики ключевых структур ГОСТа.

1. ***Ключ*** является массивом из восьми 32-битовых элементов кода, далее в настоящей работе он обозначается символом $\boldsymbol{K}$: $\boldsymbol{K} = \{\boldsymbol{K}_i\}_{0 < i \leq 7}$. В ГОСТе элементы ключа используются как 32-разрядные целые числа без знака: $0 \leq \boldsymbol{K}_i \leq 2^{32}$. Таким образом, размер ключа составляет $32 \cdot 8 = 256$ бит или 32 байта.

2. ***Таблица замен*** может быть представлена в виде матрицы размером 8×16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15. Строки ***таблицы замен*** называются ***узлами замен***, они должны содержать различные значения, то есть каждый ***узел замен*** должен содержать 16 различных чисел от 0 до 15 в произвольном порядке. Таким образом, общий объем таблицы замен равен: 8 узлов 16 элементов/узел 4 бита/элемент = 512 бит или 64 байта.

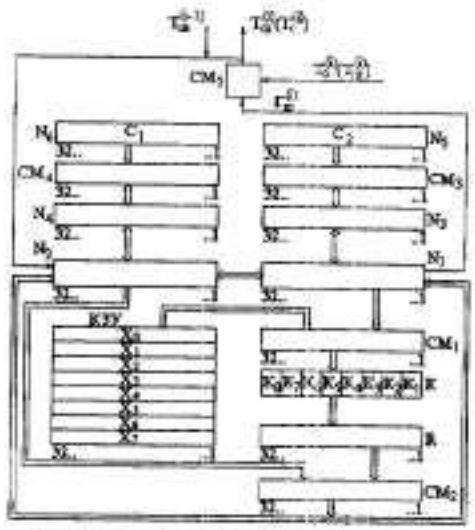


Рис. 1. Структурная схема алгоритма

Данная схема содержит:

- Четыре накопителя по 32 бита: N_1, N_2, N_3, N_4.
- Два 32-разрядных накопителя: N_5 и N_6 — с записанными в них постоянными заполнениями C_2 и C_1 соответственно.
- Ключевое запоминающее устройство (КЗУ) на 256 бит. КЗУ состоит из восьми накопителей по 32 разряда каждый: X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7.
- 32-разрядный сумматор по модулю 2: CM_2.
- Еще один сумматор по модулю 2, который не имеет ограничения на разрядность (но используется 64 бита): CM_5.
- Два сумматора по модулю 232 разрядности 32 бита: CM_1, CM_3.
- Сумматор по модулю (232-1): CM_4.
- Блок подстановки K: восемь узлов замены K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8, каждый с памятью на 64 бита.
- Регистр циклического сдвига влево на 11 бит R.

Основа алгоритма шифра — сеть Фейстеля. Выделяют четыре режима работы ГОСТ 28147-89:

1) простой замены,

2) гаммирование,

3) гаммирование с обратной связью,

4) режим выработки имитовставки.

Базовым режимом шифрования по ГОСТ 28147-89 является режим простой замены.

Для зашифрования в этом режиме открытый текст сначала разбивается на две половины (младшие биты — A, старшие биты — B. На i-м цикле используется подключ K_i:

$Ai+1=Bi \oplus f(Ai,Ki)$ ($\oplus$ – двоичное «исключающее или»)

$Bi+1=Ai$.

Для генерации подключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: $K_1 \ldots K_8$.

Ключи $K_9 \ldots K_{24}$ являются циклическим повторением ключей $K_1 \ldots K_8$ (нумеруются от младших битов к старшим). Ключи $K_{25} \ldots K_{32}$ являются ключами $K_1 \ldots K_8$, идущими в обратном порядке.

После выполнения всех 32 раундов алгоритма блоки A_{32} и B_{32} склеиваются (обратите внимание, что старшим битом становится A_{32}, а младшим — B_{32}) — результат есть результат работы алгоритма. Дешифрование выполняется так же, как и шифрование, но инвертируется порядок подключей K_i.

Функция f(A_i, Ki), используемая в сети Фейстеля

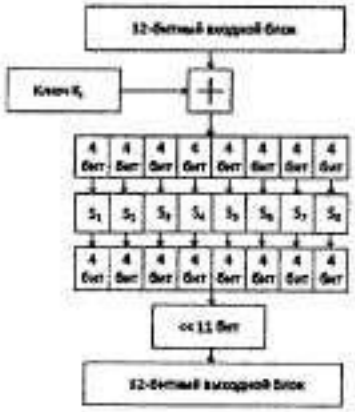


Рис. 2. Функция f(A_i, Ki), используемая в сети Фейстеля

Функция f(A_i,K_i), (рис.2), вычисляется следующим образом: A_i и K_i складываются по модулю 2^{32}.

Результат разбивается на восемь 4-битовых подпоследовательностей, каждая из которых поступает на вход своего *узла таблицы замен* (в порядке возрастания старшинства битов), называемого *S-блоком*. Общее количество S-блоков ГОСТа — восемь, то есть столько же, сколько и подпоследовательностей. Каждый *S-блок* представляет собой перестановку чисел от 0 до 15. Первая 4-битная подпоследовательность попадает на вход первого S-блока, вторая — на вход второго и т. д.

Если *S-блок* выглядит так:

1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12

и на входе S-блока 0, то на выходе будет 1, если 4, то на выходе будет 5, если на входе 12, то на выходе 6 и т. д.

Выходы всех восьми S-блоков объединяются в 32-битное слово, затем всё слово циклически сдвигается влево (к старшим разрядам) на 11 битов.

Все восемь S-блоков могут быть различными. Фактически они могут являться дополнительным ключевым материалом, но чаще являются параметром схемы, общим для определенной группы пользователей.

Режим простой замены имеет следующие недостатки:

- Может применяться только для шифрования открытых текстов с длиной, кратной 64 битам.
- При шифровании одинаковых блоков открытого текста получаются одинаковые блоки шифротекста, что может дать определенную информацию криптоаналитику.

Таким образом, применение ГОСТ 28147-89 в режиме простой замены желательно лишь для шифрования ключевых данных.

Гаммирование

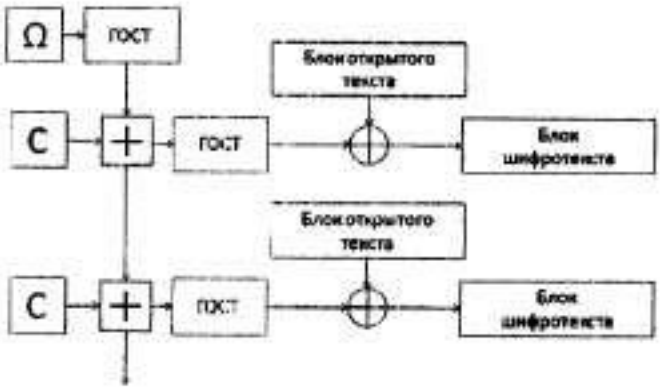


Рис.3. Схема работы в режиме гаммирования

При работе ГОСТ 28147-89 в режиме гаммирования, (рис. 3), формируется криптографическая гамма, которая затем побитно складывается по модулю 2 с исходным открытым текстом для получения шифротекста. Шифрование в режиме гаммирования лишено недостатков, присущих режиму простой замены. Даже идентичные блоки исходного текста дают разный шифротекст, а для текстов с длиной, не кратной 64 битам, "лишние" биты гаммы отбрасываются. Кроме того, гамма может быть выработана заранее, что соответствует работе шифра в поточном режиме.

Выработка гаммы происходит на основе ключа и так называемой синхропосылки, которая задает начальное состояние генератора. Алгоритм выработки следующий:

1. Синхропосылка шифруется с использованием описанного алгоритма простой замены, полученные значения записываются во вспомогательные 32-разрядные регистры N_3 и N_4 - младшие и старшие биты соответственно.
2. N_3 суммируется по модулю 2^{32} с константой $C_2 = 1010101_{16}$.
3. N_4 суммируется по модулю 2^{32}-1 с константой $C_1 = 1010104_{16}$.

4. N_3 и N_4 переписываются соответственно в N_1 и N_2, которые затем шифруются с использованием алгоритма простой замены. Полученный результат является 64 битами гаммы.
5. Шаги 2-4 повторяются в соответствии с длиной шифруемого текста.

Для расшифровывания необходимо выработать такую же гамму, после чего побитно сложить её по модулю 2 с зашифрованным текстом. Очевидно, для этого нужно использовать ту же синхропосылку, что и при шифровании. При этом, исходя из требований уникальности гаммы, нельзя использовать одну синхропосылку для шифрования нескольких массивов данных. Как правило, синхропосылка тем или иным образом передается вместе с шифротекстом.

Особенность работы ГОСТ 28147-89 в режиме гаммирования заключается в том, что при изменении одного бита шифротекста изменяется только один бит расшифрованного текста. С одной стороны, это может оказывать положительное влияние на помехозащищённость, с другой - злоумышленник может внести некоторые изменения в текст, даже не расшифровывая его.

Гаммирование с обратной связью

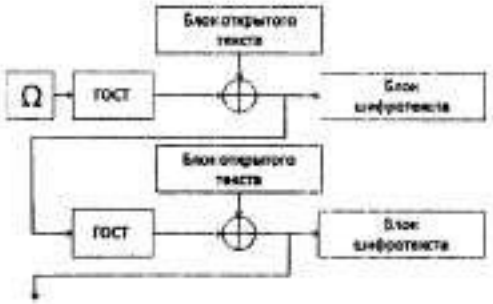


Рис. 4. Схема работы в режиме гаммирования с обратной связью

Алгоритм шифрования похож на режим гаммирования, однако гамма формируется на основе предыдущего блока зашифрованных данных, так что результат шифрования текущего блока зависит также и от предыдущих блоков. По этой причине данный режим работы также называют гаммированием с зацеплением блоков.

Алгоритм шифрования представленный на рис. 4 следующий:

1. Синхропосылка заносится в регистры N_1 и N_2.
2. Содержимое регистров N_1 и N_2 шифруется в соответствии с алгоритмом простой замены. Полученный результат является 64-битным блоком гаммы.
3. Блок гаммы побитно складывается по модулю 2 с блоком открытого текста. Полученный шифротекст заносится в регистры N_1 и N_2.
4. Операции 2-3 выполняются для оставшихся блоков требующего шифрования текста.

При изменении одного бита шифротекста, полученного с использованием алгоритма гаммирования с обратной связью, в соответствующем блоке расшифрованного текста меняется только один бит, так же затрагивается последующий блок открытого текста. При этом все остальные блоки остаются неизменными.

При использовании данного режима следует иметь в виду, что синхропосылку нельзя использовать повторно (например, при шифровании логически раздельных блоков информации - сетевых пакетов, секторов жёсткого диска и т. п). Это обусловлено тем, что первый блок шифротекста получен всего лишь сложением по модулю два с зашифрованной синхропосылкой; таким образом, знание всего лишь 8 первых байт исходного и шифрованного текста позволяет читать первые 8 байт любого другого шифротекста после повторного использования синхропосылки.

Режим выработки имитовставки

Рис. 5. Схема выработки имитовставки

Этот режим, (рис. 5), не является в общепринятом смысле режимом шифрования. При работе в режиме выработки имитовставки создаётся некоторый дополнительный блок, зависящий от всего текста и ключевых данных. Данный блок используется для проверки того, что в шифротекст случайно или преднамеренно не были внесены искажения.

Имитовставка вырабатывается для $M \geq 2$ блоков открытого текста по 64 бита.

Алгоритм следующий:

1. Блок открытых данных записывается в регистры N_1 и N_2, после чего подвергается преобразованию, соответствующему первым 16 циклам шифрования в режиме простой замены.
2. К полученному результату побитно по модулю 2 прибавляется следующий блок открытых данных. Последний блок при необходимости дополняется нулями. Сумма также шифруется в соответствии с пунктом 1.
3. После добавления и шифрования последнего блока из результата выбирается имитовставка длиной L бит: с бита номер 32-L до 32 (отсчёт начинается с 1). Стандарт рекомендует выбирать L исходя из того, что вероятность навязывания ложных данных равна 2^{-L}. Имитовставка передается по каналу связи после зашифрованных блоков.

Для проверки принимающая сторона проводит аналогичную процедуру. В случае несовпадения результата с переданной имитовставкой все соответствующие М блоков считаются ложными.
Следует отметить, что выработка имитовставки может проводиться параллельно шифрованию с использованием одного из описанных выше режимов работы.

Криптоанализ

В шифре ГОСТ 28147-89 используется 256-битовый ключ и объем ключевого пространства составляет 2^{256}. Ни на одном из существующих в настоящее время компьютеров общего применения нельзя подобрать ключ за время, меньшее многих сотен лет. Российский стандарт ГОСТ 28147-89 проектировался с большим запасом и по стойкости на много порядков превосходит американский стандарт DES с его реальным размером ключа в 56 бит и объемом ключевого пространства всего 2^{56} [4].

Существуют атаки и на полнораундовый ГОСТ 28147—89 без каких-либо модификаций. Одна из первых открытых работ, в которых был проведен анализ алгоритма, использует слабости процедуры расширения ключа ряда известных алгоритмов шифрования. В частности, полнораундовый алгоритм ГОСТ 28147—89 может быть вскрыт с помощью дифференциального криптоанализа на связанных ключах, но только в случае использования слабых таблиц замен. 24-раундовый вариант алгоритма вскрывается аналогичным образом при любых таблицах замен, однако сильные таблицы замен делают такую атаку абсолютно непрактичной.

В 2004 году группа специалистов из Кореи предложила атаку, с помощью которой, используя дифференциальный криптоанализ на связанных ключах, можно получить с вероятностью 91, 7 % 12 бит секретного ключа. Для атаки требуется 2^{35} выбранных открытых текстов и 2^{36} операций шифрования. Как видно, данная атака практически бесполезна для реального вскрытия алгоритма.

Таблица замен является долговременным ключевым элементом, то есть действует в течение гораздо более длительного срока, чем отдельный ключ. От качества этой таблицы зависит качество шифра. При "сильной" таблице замен стойкость шифра не опускается ниже некоторого допустимого предела даже в случае ее

разглашения. И наоборот, использование "слабой" таблицы может уменьшить стойкость шифра до недопустимо низкого предела.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

В данной лабораторной работе мы рассмотрим только два режима работы (простой замены и гаммирования).

1. В папке «лабораторная работа ГОСТ» представлены два режима работы. Откройте файл «режим простая замена» в программе «DEV C++».

2. После открытия создайте входной файл небольшого размера (лучше использовать текстовые выражения), запустите программу и проанализируйте, как программа зашифровала и дешифровала исходный файл. Сделайте выводы.

3. В такой же последовательности проведите 3-5 исследований для других исходных файлов. Сделайте выводы.

4. Откройте файл «режим гаммирования» в программе «DEV C++». После запуска программа запросит ввести исходные данные. В качестве них можно использовать фразы, словосочетания. Далее необходимо ввести секретный ключ (можно использовать любой набор цифр). После того как ввели секретный ключ, программа выведет на экран зашифрованные данные, затем потребует ввести секретный ключ (необходимо ввести тот же ключ, что использовали первоначально) и на экран будут выведены расшифрованные данные. Сделайте вывод.

5. Для режима гаммирования необходимо также провести 3-5 исследований для различных исходных файлов и разных секретных ключей. Сделайте вывод.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Режимы работы ГОСТ 28147-89.
2. Поясните принцип работы режима простой замены.
3. Особенность работы ГОСТ 28147-89 в режиме гаммирования.
4. Алгоритм шифрования в режиме гаммирования с обратной связью.
5. Режим выработки имитовставки.

УКАЗАНИЯ К СОСТАВЛЕНИЮ ОТЧЕТА

Отчет в электронном или бумажном виде представить преподавателю. В отчет необходимо вставить изображения с исходными, зашифрованными и дешифрованными данными. Все изображения должны быть подписаны.

Отчет о лабораторной работе должен содержать:

- цель работы;
- краткие сведения теории;
- изображения;
- выводы по пунктам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. Постановление Государственного комитета СССР по стандартам от 02.06.89. № 1409.

2. Государственный стандарт Союза ССР. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. - Издание официальное. – М.: ИПК ИЗДАТЕЛЬСТВО СТАНДАРТОВ, 1990. – 28 с.

3. ГОСТ 28147-89. – Национальная библиотека им. Н. Э. Баумана, 2016. – 15 с.

4. Басалова Г. В. Основы криптографии: учеб. пособие. – М.: ИНТУИТ, 2016. – 282 с.

5. Лидовский В. В. Основы теории информации и криптографии. – М.: ИНТУИТ, 2016. – 141 с.

6. Малюк А. А. Теория защиты информации – М.: Горячая линия-Телеком, 2015. – 184 с.

7. Каширская Е. Н. Криптографический анализ и методы защиты информации: учеб. пособие. – М.: МИРЭА, 2020. – 91 с.

8. Краковский Ю. М. Методы защиты информации. – СПб.: Лань, 2021. – 236 с.

Содержание

Защищенные системы передачи информации

Составители: К и р и л л о в Сергей Николаевич
Д м и т р и е в Владимир Тимурович

Редактор Р. К. Мангутова
Корректор С.В. Макушина
Подписано в печать 5.07.21. Формат бумаги 60x84 1/16.
Бумага писчая. Печать трафаретная. Усл. печ. л. 3,0.
Тираж: 50 экз. Заказ 3975.
Рязанский государственный радиотехнический университет.
390005, Рязань, ул. Гагарина, 59/1.
Редакционно-издательский центр РГРТУ.