Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра сетей и устройств телекоммуникаций

КРИПТОГРАФИЧЕСКОЕ КОДИРОВАНИЕ ИНФОРМАЦИИ

Методические указания
к лабораторной работе
по дисциплинам «Основы защиты информации»
и «Криптографическая защита информации в телекоммуникациях»
для студентов специальности «Сети телекоммуникаций»
дневной, вечерней и заочной форм обучения

В 3-х частях

Часть 1
ШИФРОВАНИЕ ИНФОРМАЦИИ СТАНДАРТОМ СССР

Минск 2003

УДК 621.391.2(075.8)
ББК 32.811 я 73
К8 2

Составители:
В.Ф. Голиков, А.В. Курилович

Криптографическое кодирование информации: Метод. указания к
К 82 лабораторной работе по дисциплинам «Основы защиты информации» и «Криптографическая защита информации в телекоммуникациях» для студентов специальности «Сети телекоммуникаций» дневной, вечерней и заочной форм обучения. В 3 ч. Ч. 1: Шифрование информации стандартом СССР / Сост. В.Ф. Голиков, А.В. Курилович. — Мн.: БГУИР, 2003. — 19 с.: ил.

Данные методические указания составлены в соответствии с рабочей программой курса и включают основные теоретические положения и контрольные вопросы для осмысления прочитанного материала. Часть 1 посвящена изучению стандарта СССР ГОСТ 28147-89 шифрования информации в криптосистемах симметричного типа.

УДК 621.391.2(075.8)
ББК 32.811 я 73

1. ЦЕЛЬ РАБОТЫ

Закрепление теоретических знаний по стандарту СССР шифрования информации в криптосистемах симметричного типа ГОСТ 28147-89.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Введение

В качестве официального алгоритма криптографического преобразования данных для систем обработки информации в Республике Беларусь выбран алгоритм, стандартизованный в ГОСТ 28147-89. Он предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации. Алгоритм шифрования данных представляет собой 64-битовый блочный алгоритм с 256-битовым ключом, который предусматривает четыре режима работы:

- шифрование данных в режиме простой замены;
- шифрование данных в режиме гаммирования;
- шифрование данных в режиме гаммирования с обратной связью;
- выработка имитовставки.

Основными режимами шифрования являются режимы с использованием гаммирования, однако они базируются на использовании шифрования данных в режиме простой замены.

2.2. Режим простой замены

2.2.1. Зашифровывание открытых данных в режиме простой замены

Открытые данные, подлежащие зашифровыванию, разбивают на 64-разрядные блоки T_O. Процедура зашифровывания 64-разрядного блока T_O в режиме простой замены включает 32 цикла $(j = 1, 2, \ldots, 32)$. В ключевое запоминающее устройство вводят 256 бит ключа K в виде восьми 32-разрядных подключей (чисел) K_i

$$K=K_7K_6K_5K_4K_3K_2K_1K_0.$$

Последовательность бит блока

$$T_O = (a_1(0), a_2(0), \ldots, a_{31}(0), a_{32}(0), b_1(0), b_2(0), \ldots, b_{31}(0), b_{32}(0))$$

разбивают на две последовательности по 32 бита: $b(0)$ и $a(0)$, где $b(0)$ — левые или старшие биты, $a(0)$ — правые или младшие биты.

Работа алгоритма в режиме простой замены изображена на рис. 1.

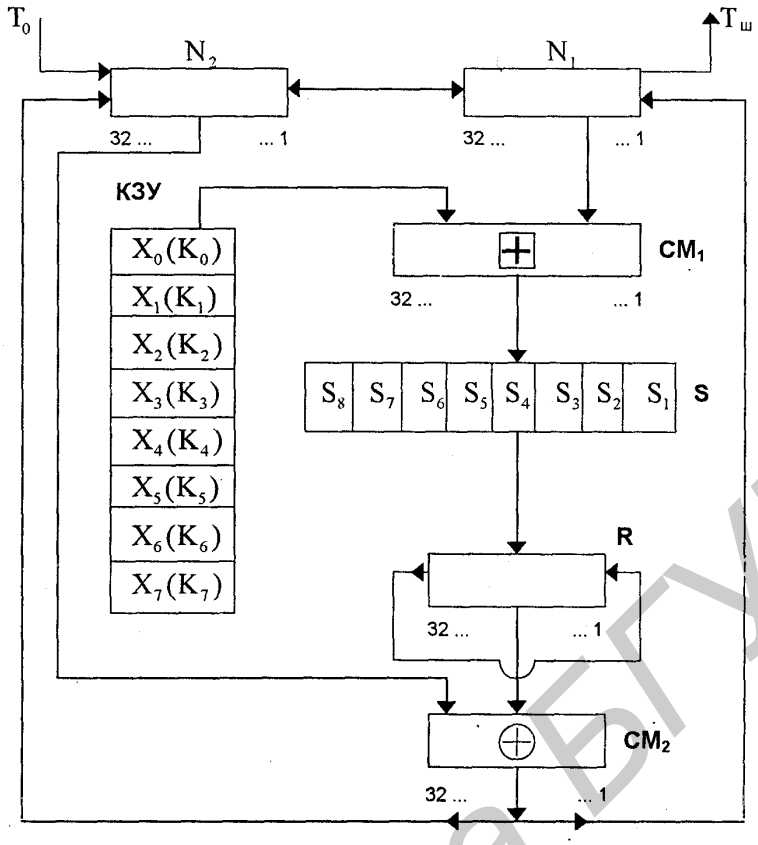

Рис. 1. Схема реализации режима простой замены

Обозначения на схеме:

N_1, N_2 — 32-разрядные накопители;

CM_1 — 32-разрядный сумматор по модулю $2^{32}(+)$;

CM_2 — 32-разрядный сумматор по модулю $2(\oplus)$;

R — 32-разрядный регистр циклического сдвига;

КЗУ — ключевое запоминающее устройство на 256 бит, состоящее из восьми 32-разрядных накопителей $X_0, X_1, X_2, \ldots, X_7$;

S — блок подстановки, состоящий из восьми узлов замены (S-блоков замены) $S_1, S_2, S_3, \ldots, S_8$.

Эти последовательности вводят в накопители N_1 и N_2 перед началом первого цикла зашифровывания. В результате начальное заполнение накопителя N_1:

$$a(0) = \begin{matrix} (a_{32}(0), & a_{31}(0), & \ldots, & a_2(0), & a_1(0)) \\ 32, & 31, & \ldots, & 2, & 1 \end{matrix} \leftarrow \text{номер разряда } N_1;$$

начальное заполнение накопителя N_2:

$$b(0) = \begin{matrix} (b_{32}(0), & b_{31}(0), & \ldots, & b_2(0), & b_1(0)) \\ 32, & 31, & \ldots, & 2, & 1 \end{matrix} \leftarrow \text{номер разряда } N_2.$$

Первый цикл $(j=1)$ процедуры зашифровывания 64-разрядного блока открытых данных можно описать уравнениями:

$$\begin{cases} a(1) = f\left(a(0) + \mathrm{K}_0\right) \oplus b(0), \\ b(1) = a(0). \end{cases}$$

Здесь $a(1)$ — заполнение N_1 после 1-го цикла зашифровывания; $b(1)$ — заполнение N_2 после 1-го цикла зашифровывания; f — функция шифрования.

Аргументом функции f является сумма по модулю 2^{32} числа $a(0)$ (начального заполнения накопителя N_1) и числа K_0 подключа, считываемого из накопителя X_0 КЗУ. Каждое из этих чисел равно 32 битам.

Функция f включает две операции над полученной 32-разрядной суммой $(a(0) + \mathrm{K}_0)$.

Первая операция называется подстановкой (заменой) и выполняется блоком подстановки S. Блок подстановки S состоит из восьми узлов замены (S-блоков замены) $\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_8$ с памятью 64 бит каждый. Поступающий из CM_1 на блок подстановки S 32-разрядный вектор разбивают на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в четырехразрядный вектор соответствующим узлом замены. Каждый узел замены можно представить в виде таблицы-перестановки шестнадцати четырехразрядных двоичных чисел в диапазоне 0000…1111. Входной вектор указывает адрес строки в таблице, а число в этой строке является выходным вектором. Затем четырехразрядные выходные векторы последовательно объединяют в 32-разрядный вектор. Узлы замены (таблицы-перестановки) представляют собой ключевые элементы, которые являются общими для сетей ТКС и редко изменяются. Эти узлы замены должны сохраняться в секрете.

Вторая операция — циклический сдвиг влево (на 11 разрядов) 32-разрядного вектора, полученного с выхода блока подстановки S. Циклический сдвиг выполняется регистром сдвига R. Затем результат работы функции шифрования f суммируют поразрядно по модулю 2 в сумматоре CM_2 с 32-разрядным начальным заполнением $b(0)$ накопителя N_2. Далее полученный на выходе CM_2 результат (значение $a(1)$) записывают в накопитель N_1, а старое значение N_1 (значение $a(0)$) переписывают в накопитель N_2 (значение $b(1) = a(0)$). Первый цикл завершен. Последующие циклы осуществляются аналогично, при этом во втором цикле из КЗУ считывают заполнение X_1 — подключ K_1, в третьем цикле — подключ K_2 и т.д., в восьмом цикле — подключ K_7. В циклах с 9-го по 16-й, а также в циклах с 17-го по 24-й подключи из КЗУ считываются в том же порядке: $\mathrm{K}_0, \mathrm{K}_1, \mathrm{K}_2, \ldots, \mathrm{K}_6, \mathrm{K}_7$. В

последних восьми циклах с 25-го по 32-й порядок считывания подключей из КЗУ обратный: $K_7, K_6, \ldots, K_2, K_1, K_0$. Таким образом, при зашифровывании в 32 циклах осуществляется следующий порядок выборки из КЗУ подключей:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7,$$

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

В 32-м цикле результат из сумматора CM_2 вводится в накопитель N_2, а в накопителе N_1 сохраняется прежнее заполнение. Полученные после 32-го цикла зашифровывания заполнения накопителей N_1 и N_2 являются блоком зашифрованных данных $T_Ш$, соответствующим блоку открытых данных T_O.

Уравнения зашифровывания в режиме простой замены имеют вид

$$\begin{cases} a(j) = f\left(a(j-1) + K_{(j-1) \bmod 8}\right) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 1\ldots24,$$

$$\begin{cases} a(j) = f\left(a(j-1) + K_{(32-j) \bmod 8}\right) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 25\ldots31,$$

$$\begin{cases} a(32) = a(31) \\ b(32) = f\left(a(31) + K_0\right) \oplus b(31) \end{cases} \quad \text{при } j = 32,$$

где $a(j) = \left(a_{32}(j), a_{31}(j), \ldots, a_1(j)\right)$ — заполнение N_1 после j-го цикла зашифровывания; $b(j) = \left(b_{32}(j), b_{31}(j), \ldots, b_1(j)\right)$ — заполнение N_2 после j-го цикла зашифровывания, $j = 1\ldots32$.

Блок зашифрованных данных $T_Ш$ (64 разряда) выводится из накопителей N_1, N_2 в следующем порядке: из разрядов 1…32 накопителя N_1, затем из разрядов 1…32 накопителя N_2, т.е. начиная с младших разрядов:

$$T_Ш = \left(a_1(32), a_2(32), \ldots, a_{31}(32), a_{32}(32), b_1(32), b_2(32), \ldots, b_{31}(32), b_{32}(32)\right).$$

Остальные блоки открытых данных зашифровываются в режиме простой замены аналогично.

2.2.2. Расшифровывание в режиме простой замены

Криптосхема, реализующая алгоритм расшифровывания в режиме простой замены, имеет тот же вид, что и при зашифровывании (см. рис. 1).

В КЗУ вводят 256 бит ключа, на котором осуществлялось зашифровывание. Зашифрованные данные, подлежащие расшифровыванию, разбиты на блоки $T_Ш$, по 64 бита в каждом. Ввод любого блока

$$T_Ш = \left(a_1(32), a_2(32), \ldots, a_{31}(32), a_{32}(32), b_1(32), b_2(32), \ldots, b_{31}(32), b_{32}(32)\right)$$

в накопители N_1 и N_2 производят так, чтобы начальное значение накопителя N_1 имело вид

$$\begin{array}{cccccc} (a_{32}(32), & a_{31}(32), & \ldots, & a_2(32), & a_1(32)) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \text{номер разряда } N_1, \end{array}$$

а начальное заполнение накопителя N_2:

$$\begin{array}{cccccc} (b_{32}(32), & b_{31}(32), & \ldots, & b_2(32), & b_1(32)) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \textit{номер разряда } N_2. \end{array}$$

Расшифровывание осуществляется по тому же алгоритму, что и зашифровывание, с тем изменением, что заполнения накопителей $X_0, X_1, X_2, \ldots, X_7$ считываются из КЗУ в циклах расшифровывания в следующем порядке:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0,$$

$$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

Уравнения расшифровывания имеют вид

$$\begin{cases} a(32-j) = f\left(a(32-j+1) + K_{j-1}\right) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 1\ldots 8,$$

$$\begin{cases} a(32-j) = f\left(a(32-j+1) + K_{(32-j) \bmod 8}\right) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 9\ldots 31,$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f\left(a(1) + K_0\right) \oplus b(1) \end{cases} \quad \text{при } j = 32.$$

Полученные после 32 циклов работы заполнения накопителей N_1 и N_2 образуют блок открытых данных

$$T_O = (a_1(0), a_2(0), \ldots, a_{31}(0), a_{32}(0), b_1(0), b_2(0), \ldots, b_{31}(0), b_{32}(0)),$$

соответствующий блоку зашифрованных данных $T_Ш$. При этом состояние накопителя N_1:

$$\begin{array}{cccccc} (a_{32}(0), & a_{31}(0), & \ldots, & a_2(0), & a_1(0)) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \text{номер разряда } N_1, \end{array}$$

состояние накопителя N_2:

$$\begin{array}{cccccc} (b_{32}(0), & b_{31}(0), & \ldots, & b_2(0), & b_1(0)) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \text{номер разряда } N_2. \end{array}$$

Аналогично расшифровываются остальные блоки зашифрованных данных.

Если алгоритм зашифровывания в режиме простой замены 64-битового блока T_O обозначить через A, то

$$A(T_O) = A(a(0), b(0)) = (a(32), b(32)) = T_Ш.$$

Следует иметь в виду, что режим простой замены допустимо использовать для шифрования данных только в ограниченных случаях при выработке ключа и зашифровывании его с обеспечением имитозащиты для передачи по каналам связи или для хранения в памяти ЭВМ.

2.3. Режим гаммирования

2.3.1. Зашифровывание открытых данных в режиме гаммирования

Криптосхема, реализующая алгоритм зашифровывания в режиме гаммирования, показана на рис. 2. Открытые данные разбивают на 64-разрядные блоки

$$\mathrm{T}_{\mathrm{O}}^{(1)}, \mathrm{T}_{\mathrm{O}}^{(2)}, \ldots, \mathrm{T}_{\mathrm{O}}^{(i)}, \ldots, \mathrm{T}_{\mathrm{O}}^{(m)},$$

где $\mathrm{T}_{\mathrm{O}}^{(i)}$ — i-й 64-разрядный блок открытых данных, $i = 1 \ldots m$, m определяется объемом шифруемых данных.

Эти блоки поочередно зашифровываются в режиме гаммирования путем поразрядного сложения по модулю 2 в сумматоре CM_5 с гаммой шифра $\Gamma_{\mathrm{Ш}}$, которая вырабатывается блоками по 64 бита, т.е.

$$\Gamma_{\mathrm{Ш}} = \left(\Gamma_{\mathrm{Ш}}^{(1)}, \Gamma_{\mathrm{Ш}}^{(2)}, \ldots, \Gamma_{\mathrm{Ш}}^{(i)}, \ldots, \Gamma_{\mathrm{Ш}}^{(m)}\right),$$

где $\Gamma_{\mathrm{Ш}}^{(i)}$ — i-й 64-разрядный блок, $i = 1 \ldots m$.

Число двоичных разрядов в блоке $\mathrm{T}_{\mathrm{O}}^{(m)}$ может быть меньше 64, при этом не использованная для зашифровывания часть гаммы шифра из блока $\Gamma_{\mathrm{Ш}}^{(m)}$ отбрасывается.

Уравнение зашифровывания данных в режиме гаммирования имеет вид

$$\mathrm{T}_{\mathrm{Ш}}^{(i)} = \mathrm{T}_{\mathrm{O}}^{(i)} \oplus \Gamma_{\mathrm{Ш}}^{(i)},$$

где $\mathrm{T}_{\mathrm{Ш}}^{(i)}$ — i-й блок 64-разрядного блока зашифрованного текста; $\Gamma_{\mathrm{Ш}}^{(i)} = \mathrm{A}\left(\mathrm{Y}_{i-1} + \mathrm{C}_2, \mathrm{Z}_{i-1} + \mathrm{C}_1\right)$, $i = 1 \ldots m$; $\mathrm{A}(*)$ — функция зашифровывания в режиме простой замены; $\mathrm{C}_1, \mathrm{C}_2$ — 32-разрядные двоичные константы; $\mathrm{Y}_i, \mathrm{Z}_i$ — 32-разрядные двоичные последовательности.

Величины $\mathrm{Y}_i, \mathrm{Z}_i$ определяются итерационно по мере формирования гаммы $\Gamma_{\mathrm{Ш}}$ следующим образом:

$$\left(\mathrm{Y}_0, \mathrm{Z}_0\right) = \mathrm{A}\left(\tilde{\mathrm{S}}\right),$$

где $\tilde{\mathrm{S}}$ — синхропосылка (64-разрядная двоичная последовательность),

$$\left(\mathrm{Y}_i, \mathrm{Z}_i\right) = \left(\mathrm{Y}_{i-1} + \mathrm{C}_2, \mathrm{Z}_{i-1} + \mathrm{C}_1\right), i = 1 \ldots m.$$

Рассмотрим реализацию процедуры зашифровывания в режиме гаммирования. В накопители N_6 и N_5 заранее записаны 32-разрядные двоичные константы C_1 и C_2, имеющие следующие значения (в шестнадцатеричной форме):

$$C_1 = 01010104_{(16)}, C_2 = 01010101_{(16)}.$$

В КЗУ вводится 256 бит ключа, в накопители N_1 и N_2 — 64-разрядная двоичная последовательность (синхропосылка)

$$\tilde{S}=(S_1, S_2, \ldots, S_{64}).$$

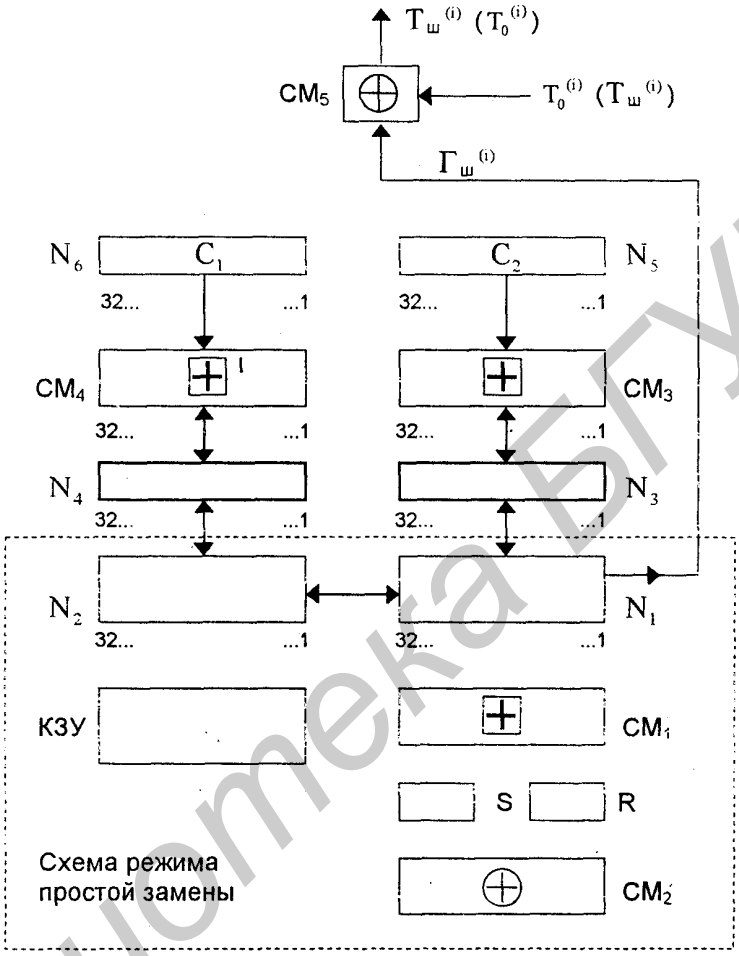

Рис. 2. Схема реализации режима гаммирования

Синхропосылка $\tilde{S}$ является исходным заполнением накопителей N_1 и N_2 для последовательной выработки m блоков гаммы шифра.

Исходное заполнение накопителя N_1:

$$(S_{32},\ S_{31},\ \ldots,\ S_2,\ S_1)$$
$$32,\ 31,\ \ldots,\ 2,\ 1 \quad \leftarrow \text{номер разряда } N_1,$$

состояние накопителя N_2:

$$(S_{64},\ S_{63},\ \ldots,\ S_{34},\ S_{33})$$
$$64,\ 63,\ \ldots,\ 34,\ 33 \quad \leftarrow \text{номер разряда } N_2.$$

Исходное заполнение N_1 и N_2 (синхропосылка $\tilde{S}$) зашифровывается в режиме простой замены. Результат зашифровывания

$$A\left(\tilde{S}\right)=\left(Y_0,Z_0\right)$$

переписывается в 32-разрядные накопители N_3 и N_4 так, что заполнение N_1 переписывается в N_3, а заполнение N_2 — в N_4.

Заполнение накопителя N_4 суммируют по модулю $2^{32}-1$ в сумматоре CM_4 с 32-разрядной константой C_1 из накопителя N_6. Результат записывается в N_4. Заполнение накопителя N_3 суммируется по модулю 2^{32} в сумматоре CM_3 с 32-разрядной константой C_3 из накопителя N_5. Результат записывается в N_3. Заполнение N_3 переписывают в N_1, а заполнение N_4 — в N_2, при этом заполнения N_3, N_4 сохраняются. Заполнение накопителей зашифровывается в режиме простой замены.

Полученное в результате зашифровывания заполнение накопителей N_1 и N_2 образует первый 64-разрядный блок гаммы шифра:

$$\Gamma_{Ш}^{(1)}=\left(\gamma_1^{(1)},\gamma_2^{(1)},...,\gamma_{63}^{(1)},\gamma_{64}^{(1)}\right),$$

который суммируют поразрядно по модулю 2 в сумматоре CM_5 с первым 64-разрядным блоком открытых данных:

$$T_{O}^{(1)}=\left(t_1^{(1)},t_2^{(1)},\ldots,t_{63}^{(1)},t_{64}^{(1)}\right).$$

В результате суммирования по модулю 2 значений $\Gamma_{Ш}^{(1)}$ и $T_{O}^{(1)}$ получают первый 64-разрядный блок зашифрованных данных

$$T_{Ш}^{(1)}=\Gamma_{Ш}^{(1)}\oplus T_{O}^{(1)}=\left(\tau_1^{(1)},\tau_2^{(1)},\ldots,\tau_{63}^{(1)},\tau_{64}^{(1)}\right),$$

где $\tau_i^{(1)}=t_i^{(1)}\oplus\gamma_i^{(1)},\ i=1\ldots64$.

Для получения следующего 64-разрядного блока гаммы шифра $\Gamma_{Ш}^{(2)}$ заполнение N_4 суммируется по модулю $\left(2^{32}-1\right)$ в сумматоре CM_4 с константой C_1 из N_6. Результат записывается в N_4. Заполнение N_3 суммируется по модулю 2^{32} в сумматоре CM_3 с константой C_2 из N_5. Результат записывается в N_3. Новое заполнение N_3 переписывают в N_1, а новое заполнение N_4 — в N_2, при этом заполнения N_3 и N_2 сохраняют. Заполнения N_1, N_2 зашифровывают в режиме простой замены.

Полученное в результате зашифровывания заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре CM_5 со вторым блоком открытых данных $T_{O}^{(2)}$:

$$T_{Ш}^{(2)}=\Gamma_{Ш}^{(2)}\oplus T_{O}^{(2)}.$$

Аналогично вырабатываются блоки гаммы шифра $\Gamma_{Ш}^{(3)},\Gamma_{Ш}^{(4)},\ldots,\Gamma_{Ш}^{(m)}$ и зашифровываются блоки открытых данных $T_{O}^{(3)},T_{O}^{(3)},\ldots,T_{O}^{(m)}$.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных:

$$T_{Ш}^{(1)}, T_{Ш}^{(2)}, \ldots, T_{Ш}^{(m)}.$$

2.3.2. Расшифровывание в режиме гаммирования

При расшифровывании криптосхема имеет тот же вид, что и при зашифровывании (см. рис. 2).

Уравнение расшифровывания

$$T_{О}^{(i)} = T_{Ш}^{(i)} \oplus \Gamma_{Ш}^{(i)} = T_{Ш}^{(i)} \oplus A\left(Y_{i-1} + C_2, Z_{i-1} + C_1\right),\ i = 1 \ldots m.$$

Следует отметить, что расшифрование данных возможно только при наличии синхропосылки, которая не является секретным элементом шифра и может храниться в памяти ЭВМ или передаваться по каналам связи вместе с зашифрованными данными.

Рассмотрим реализацию процедуры расшифровывания. В КЗУ вводят 256 бит ключа, с помощью которого осуществляется зашифровывание данных $T_{О}^{(1)}, T_{О}^{(2)}, \ldots, T_{О}^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка и осуществляется процесс выработки m блоков гаммы шифра $\Gamma_{Ш}^{(1)}, \Gamma_{Ш}^{(2)}, \ldots, \Gamma_{Ш}^{(m)}$. Блоки зашифрованных данных $T_{Ш}^{(1)}, T_{Ш}^{(2)}, \ldots, T_{Ш}^{(m)}$ суммируются поразрядно по модулю 2 в сумматоре $СМ_5$ с блоками гаммы шифра $\Gamma_{Ш}^{(1)}, \Gamma_{Ш}^{(2)}, \ldots, \Gamma_{Ш}^{(m)}$. В результате получаются блоки открытых данных $T_{О}^{(1)}, T_{О}^{(2)}, \ldots, T_{О}^{(m)}$, при этом $T_{О}^{(m)}$ может содержать меньше 64 разрядов.

2.4. Режим гаммирования с обратной связью

Криптосхема, реализующая алгоритм зашифровывания в режиме гаммирования с обратной связью, имеет вид, показанный на рис. 3.

2.4.1. Зашифровывание открытых данных в режиме гаммирования с обратной связью

Открытые данные, разбитые на 64-разрядные блоки $T_{О}^{(1)}, T_{О}^{(2)}, \ldots, T_{О}^{(m)}$, зашифровываются в режиме гаммирования с обратной связью путем поразрядного сложения по модулю 2 с гаммой шифра $\Gamma_{Ш}$, которая вырабатывается блоками по 64 бита: $\Gamma_{Ш}^{(1)}, \Gamma_{Ш}^{(2)}, \ldots, \Gamma_{Ш}^{(m)}$.

Число двоичных разрядов в блоке $T_{О}^{(m)}$ может быть меньше 64, при этом не использованная для шифрования часть гаммы шифра из блока $\Gamma_{Ш}^{(m)}$ отбрасывается.

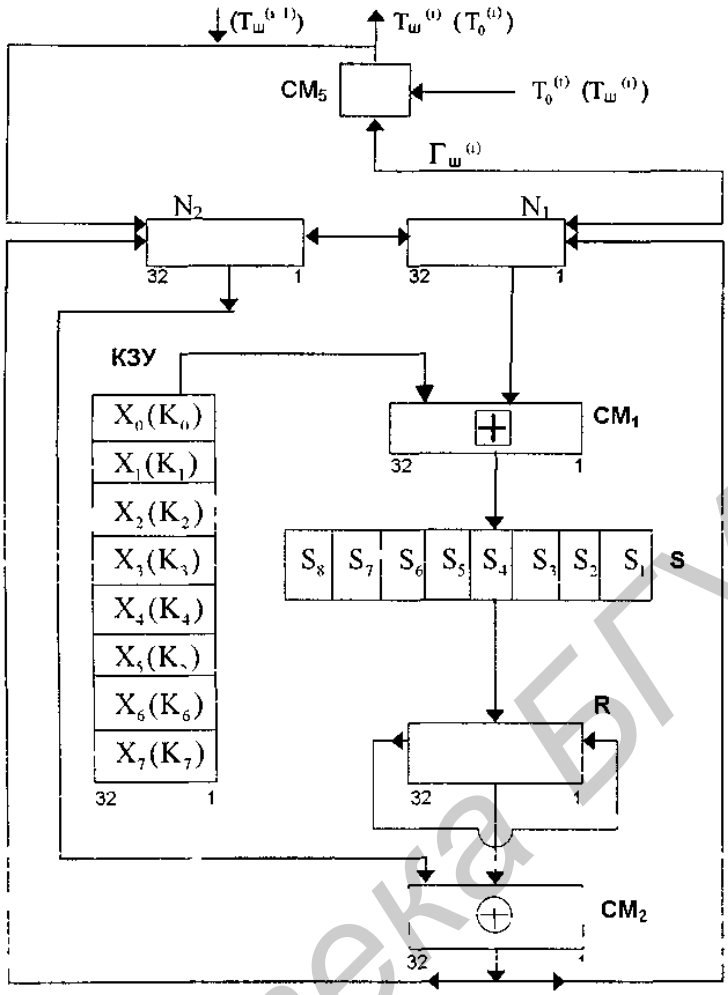

Рис. 3. Схема реализации режима гаммирования с обратной связью

Уравнения зашифровывания в режиме гаммирования с обратной связью имеют вид:

$$\mathrm{T}_{\mathrm{Ш}}^{(1)} = \mathrm{A}\left(\tilde{\mathrm{S}}\right) \oplus \mathrm{T}_{\mathrm{O}}^{(1)} = \Gamma_{\mathrm{Ш}}^{(1)} \oplus \mathrm{T}_{\mathrm{O}}^{(1)},$$

$$\mathrm{T}_{\mathrm{Ш}}^{(i)} = \mathrm{A}\left(\mathrm{T}_{\mathrm{Ш}}^{(i-1)}\right) \oplus \mathrm{T}_{\mathrm{O}}^{(i)} = \Gamma_{\mathrm{Ш}}^{(i)} \oplus \mathrm{T}_{\mathrm{O}}^{(i)},\ i = 2 \ldots m.$$

Здесь $\mathrm{T}_{\mathrm{Ш}}^{(i)}$ — i-й 64-разрядный блок зашифрованного текста; $\mathrm{A}(*)$ — функция зашифровывания в режиме простой замены; m определяется объемом открытых данных.

Аргументом функции $\mathrm{A}(*)$ на первом шаге итеративного алгоритма является 64-разрядная синхропосылка S, а на всех последующих шагах — предыдущий блок зашифрованных данных $\mathrm{T}_{\mathrm{Ш}}^{(i-1)}$.

Процедура зашифровывания данных в режиме гаммирования с обратной связью реализуется следующим образом. В КЗУ вводятся 256 бит ключа, в накопители N_1 и N_2 вводится синхропосылка $\tilde{\mathrm{S}}=(\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_{64})$ из 64 бит. Исходное заполнение накопителей N_1 и N_2 зашифровывается в режиме

простой замены. Полученное в результате зашифровывания заполнение накопителей N_1 и N_2 образует первый 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(1)} = A\left(\tilde{S}\right)$, который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ с первым 64-разрядным блоком открытых данных

$$T_{О}^{(1)} = \left(t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)}\right).$$

В результате получают первый 64-разрядный блок зашифрованных данных

$$T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{О}^{(1)},$$

где $T_{Ш}^{(1)} = \left(\tau_1^{(1)}, \tau_2^{(1)}, \ldots, \tau_{63}^{(1)}, \tau_{64}^{(1)}\right)$.

Блок зашифрованных данных $T_{Ш}^{(i)}$ одновременно является также исходным состоянием накопителей N_1 и N_2 для выработки второго блока гаммы шифра $\Gamma_{Ш}^{(2)}$, и поэтому по обратной связи $T_{Ш}^{(1)}$ записывается в указанные накопители N_1 и N_2.

Заполнение накопителя N_1:

$$\left(\tau_{32}^{(1)}, \quad \tau_{31}^{(1)}, \quad \ldots, \quad \tau_2^{(1)}, \quad \tau_1^{(1)}\right)$$

$$32, \quad 31, \quad \ldots, \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_1.$$

Заполнение накопителя N_2:

$$\left(\tau_{64}^{(1)}, \quad \tau_{63}^{(1)}, \quad \ldots, \quad \tau_{34}^{(1)}, \quad \tau_{33}^{(1)}\right)$$

$$32, \quad 31, \quad \ldots, \quad 2, \quad 1 \quad \leftarrow \text{номер разряда } N_1.$$

Заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены. Полученное в результате зашифровывания заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре СМ со вторым блоком открытых данных $T_{О}^{(2)}$:

$$T_{Ш}^{(2)} = \Gamma_{Ш}^{(2)} \oplus T_{О}^{(2)}.$$

Выработка последующих блоков гаммы шифра $\Gamma_{Ш}$ и зашифровывание соответствующих блоков открытых данных $T_{О}^{(i)}$ $(i = 3 \ldots m)$ производятся аналогично. Если длина последнего m-го блока открытых данных $T_{О}^{(m)}$ меньше 64 разрядов, то из $\Gamma_{Ш}^{(m)}$ используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных $T_{Ш}^{(1)}, T_{Ш}^{(2)}, \ldots, T_{Ш}^{(m)}$.

2.4.2. Расшифровывание в режиме гаммирования с обратной связью

При расшифровывании криптосхема имеет тот же вид, что и при зашифровывании (см. рис. 3).

Уравнения расшифровывания:

$$T_{О}^{(1)} = A\left(\tilde{S}\right) \oplus T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{Ш}^{(1)},$$

$$T_{О}^{(i)} = \Gamma_{Ш}^{(i)} \oplus T_{Ш}^{(i)} = A\left(T_{Ш}^{(i-1)}\right) \oplus T_{Ш}^{(i)},\ i = 2 \ldots m.$$

Реализация процедуры расшифровывания зашифрованных данных в режиме гаммирования с обратной связью происходит следующим образом. В КЗУ вводят 256 бит того же ключа, на котором осуществлялось зашифровывание открытых блоков $T_{О}^{(1)}, T_{О}^{(2)}, \ldots, T_{О}^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка $\tilde{S}$. Исходное заполнение накопителей N_1 и N_2 (синхропосылка $\tilde{S}$) зашифровывается в режиме простой замены. Полученное в результате зашифровывания заполнение N_1 и N_2 образует первый блок гаммы шифра

$$\Gamma_{Ш}^{(1)} = A\left(\tilde{S}\right),$$

который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ с блоком зашифрованных данных $T_{Ш}^{(1)}$. В результате получается первый блок открытых данных

$$T_{О}^{(2)} = \Gamma_{Ш}^{(2)} \oplus T_{Ш}^{(2)}.$$

Блок зашифрованных данных $T_{Ш}^{(1)}$ является исходным заполнением накопителей N_1 и N_2 для выработки второго блока гаммы шифра $\Gamma_{Ш}^{(2)}$: $\Gamma_{Ш}^{(2)} = A\left(T_{Ш}^{(1)}\right)$. Полученное заполнение накопителей N_1 и N_2 зашифровывается в режиме простой замены. Образованный в результате зашифровывания блок $\Gamma_{Ш}^{(2)}$ суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ со вторым блоком зашифрованных данных $T_{Ш}^{(2)}$. В результате получают второй блок открытых данных. Аналогично в N_1 и N_2 последовательно записывают блоки зашифрованных данных $T_{Ш}^{(2)}, T_{Ш}^{(3)}, \ldots, T_{Ш}^{(m)}$, из которых в режиме простой замены вырабатываются блоки гаммы шифра $\Gamma_{Ш}^{(3)}, \Gamma_{Ш}^{(4)}, \ldots, \Gamma_{Ш}^{(m)}$. Блоки гаммы шифра суммируются поразрядно по модулю 2 в сумматоре $СМ_5$ с блоками зашифрованных данных $T_{Ш}^{(3)}, T_{Ш}^{(4)}, \ldots, T_{Ш}^{(m)}$.

В результате получают блоки открытых данных $T_{О}^{(3)}, T_{О}^{(4)}, \ldots, T_{О}^{(m)}$, при этом последний блок открытых данных $T_{О}^{(m)}$ может содержать меньше 64 разрядов.

2.5. Режим выработки имитовставки

Имитовставка — это блок из P бит, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к зашифрованным данным для обеспечения их имитозащиты.

Имитозащита — это защита системы шифрованной связи от навязывания ложных данных.

В стандарте ГОСТ 28147-89 определяется процесс выработки имитовставки, который единообразен для любого из режимов шифрования данных. Имитовставка $И_P$ вырабатывается из блоков открытых данных либо перед шифрованием всего сообщения, либо параллельно с шифрованием по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (например, адресную часть, время, синхропосылку) и не зашифровываются.

Значение параметра P (число двоичных разрядов в имитовставке) определяется криптографическими требованиями с учетом того, что вероятность навязывания ложных помех равна $1/2^P$.

Для выработки имитовставки открытые данные представляют в виде последовательности 64-разрядных блоков $T_O^{(i)}$, $i = 1 \ldots m$. Первый блок открытых данных $T_O^{(1)}$ подвергают преобразованию $\tilde{A}(*)$, соответствующему первым 16 циклам алгоритма зашифрования в режиме простой замены. В качестве ключа для выработки имитовставки используют ключ длиной 256 бит, по которому шифруют данные.

Полученное после 16 циклов 64-разрядное число $\tilde{A}\left(T_O^{(1)}\right)$ суммируют по модулю 2 со вторым блоком открытых данных $T_O^{(2)}$. Результат суммирования $\tilde{A}\left(T_O^{(1)}\right) \oplus T_O^{(2)}$ снова подвергают преобразованию $\tilde{A}(*)$.

Полученное 64-разрядное число $\tilde{A}\left(\tilde{A}\left(T_O^{(1)}\right) \oplus T_O^{(2)}\right)$ суммируют по модулю 2 с третьим блоком $T_O^{(3)}$ и снова подвергают преобразованию $\tilde{A}(*)$, получая 64-разрядное число $\tilde{A}\left(\tilde{A}\left(\tilde{A}\left(T_O^{(1)}\right) \oplus T_O^{(2)}\right) \oplus T_O^{(3)}\right)$, и т.д.

Последний блок $T_O^{(m)}$ (при необходимости дополненный нулями до полного 64-разрядного блока) суммируют по модулю 2 с результатом вычислений на шаге $(m-1)$, после чего зашифровывают в режиме простой замены, используя преобразование $\tilde{A}(*)$.

Из полученного 64-разрядного числа выбирают отрезок $И_P$ (имитовставку) длиной P бит:

$$И_P = \left[a_{32-P+1}^m(16), a_{32-P+2}^m(16), \ldots, a_{32}^m(16) \right],$$

где $a_i^{(m)}$ — i бит 64-разрядного числа, полученного после 16-го цикла последнего преобразования $\tilde{A}(*)$, $32-P+1 \le i \le 31$.

Имитовставка $И_P$ передается по каналу связи в конце зашифрованных данных, т.е.

$$Т_Ш^{(1)}, Т_Ш^{(2)}, \ldots, Т_Ш^{(m)}, И_P.$$

Поступившие к получателю зашифрованные данные $Т_Ш^{(1)}, Т_Ш^{(2)}, \ldots, Т_Ш^{(m)}$ расшифровываются, и из полученных блоков открытых данных $Т_О^{(1)}, Т_О^{(2)}, \ldots, Т_О^{(m)}$ аналогичным образом вырабатывается имитовставка $И'_P$, которая сравнивается с $И_P$. В случае несовпадения блок открытых данных считается ложным.

3. ПРЕДВАРИТЕЛЬНОЕ ЗАДАНИЕ

3.1.Изучите теоретическую часть.

3.2.Найдите сумму по модулю 2 двух чисел 2940553835_{10} и 3984555948_{10}.

3.3.Найдите сумму по модулю 3 двух чисел 3496_{10} и 3718_{10}.

3.4.Найдите сумму по модулю 2^{32} следующих пар чисел: 3037741847_{10} и 1257225448_{10}, 2706981523_{10} и 1587985773_{10}, 2597745569_{10} и 1697221728_{10}.

3.5.Пусть каждая буква из 16 первых букв русского алфавита (абвгдежзийклмноп) имеет четырехразрядный двоичный код, соответствующий ее номеру от 0 до 15, т.е. а — 0000_2, б — 0001_2, …, п — 1111_2. Составьте из этих букв произвольное сообщение из 32 символов, затем разбейте полученное сообщение на блоки длиной соответствующей разрядности накопителей N_1 и N_2. Значения полученных блоков запишите в десятичной системе счисления.

3.6.Найдите состояние 32-разрядного двоичного регистра сдвига после циклического сдвига вправо на 3 числа 298865410_{10}, предварительно записанного в регистр.

4.ЛАБОРАТОРНОЕ ЗАДАНИЕ

4.1.Включите ПЭВМ.

4.2.Запустите программу gost.exe на выполнение. Данная программа реализует алгоритм зашифрования и расшифрования данных по стандарту ГОСТ 28147-89 в режиме простой замены. Входные и выходные данные работы этого алгоритма представляют собой 64-разрядные двоичные числа, поэтому, если данные, подлежащие обработке, представлены в другом виде, их предварительно переводят в двоичный вид и разбивают на блоки длиной по 64 бита. Для упрощения работы с 64-разрядными двоичными числами блоки представляются в виде двух блоков по 32 бита (соответствующих

разрядности накопителей N_1 и N_2) и записываются в десятичной системе счисления.

4.3.Нажмите кнопку «Начать выполнение работы». Выполнение работы состоит в том, что Вы должны вручную найти значения в контрольных точках работы алгоритма. Для этого необходимо ввести полученное значение в соответствующее поле рабочего окна программы и нажать кнопку «Проверить». Если расчет выполнен правильно, происходит переход к следующей контрольной точке.

4.4.Зашифровывание в режиме простой замены.

4.4.1.Введите (в десятичной системе счисления) начальное состояние накопителей N_1 и N_2.

4.4.2.Руководствуйтесь инструкциями в рабочем окне программы.

4.4.3.Для того чтобы скопировать в буфер обмена результат какой-либо контрольной точки, достаточно один раз щелкнуть по нему. Подтверждением того, что копирование произошло, является кратковременное изменение цветового фона.

4.5.Расшифровывание в режиме простой замены.

4.5.1.Введите (в десятичной системе счисления) начальное состояние накопителей N_1 и N_2.

4.5.2.Руководствуйтесь инструкциями в рабочем окне программы.

3.3.Оформите отчет и сделайте выводы.

5.СОДЕРЖАНИЕ ОТЧЕТА

5.1.Решение задач предварительного задания.

5.2.Результаты выполнения работы.

5.3.Анализ результатов и выводы.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

6.1. Какие существуют режимы работы алгоритма?

6.2. К какому типу криптосистем относится алгоритм?

6.3. Какой разрядности ключ используется в алгоритме?

6.4. Поясните принцип работы блока подстановки.

6.5. Перечислите основные достоинства и недостатки алгоритма.

6.6. Какое максимальное количество двоичных разрядов может содержать результат при вычислениях по модулям 2, 8, 9, 2^{32}?

6.7. Могут ли совпадать имитовставки различных сообщений?

6.8. Как повлияет искажение одного бита шифротекста на передаваемую информацию при разных режимах работы алгоритма?

ЛИТЕРАТУРА

Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. М.: Радио и связь, 1999.

Мельников В.В. Защита информации в компьютерных системах. М.: Финансы и статистика, 1997.

Харин Ю.С., Берник В.И., Матвеев Г.В. Математические основы криптологии. Мн.: БГУ, 1999.

Герасименко В.А., Малюк А.А. Основы защиты информации. М.: МГИФИ, 1997.

Леонов А.П., Леонов К.П., Фролов Г.В. Безопасность автоматизированных банковских и офисных технологий. Мн.: Нац. кн. палата Беларуси, 1996.

Зима В.М., Молдовян А.А., Молдовян Н.А. Компьютерные сети и защита передаваемой информации. СПб.: СПбГУ, 1998.

Св. план 2003, поз. 89

Учебное издание

КРИПТОГРАФИЧЕСКОЕ КОДИРОВАНИЕ ИНФОРМАЦИИ

Методические указания
к лабораторной работе
по дисциплинам «Основы защиты информации»
и «Криптографическая защита информации в телекоммуникациях»
для студентов специальности «Сети телекоммуникаций»
дневной, вечерней и заочной форм обучения

В 3-х частях

Часть 1

ШИФРОВАНИЕ ИНФОРМАЦИИ СТАНДАРТОМ СССР

Составители:
Голиков Владимир Федорович,
Курилович Андрей Владимирович

Редактор Н.А. Бебель
Корректор Е.Н. Батурчик

Подписано в печать. Формат 60×84 1/16.
Бумага офсетная. Печать ризографическая. Гарнитура «Таймс». Усл. печ. л.
Уч.-изд. л. 0,7. Тираж 100 экз. Заказ 607.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия ЛП № 156 от 05.02. 2001.
Лицензия ЛВ № 509 от 03.08. 2001.
220013, Минск, П. Бровки, 6.