ДЕРЖАВНИЙ СТАНДАРТ УКРАЇНИ

Система обробки інформації.

Криптографічний захист інформації.

Алгоритм криптографічного перетворення (ГОСТ 28147-89)

ДСТУ ГОСТ 28147 - 2009

Видання офіційне

КИЇВ

ДЕРЖСТАНДАРТ УКРАЇНИ

2009

ОКС 35.040, 35.060, 35.080, 35.100, 35.140, 35.240 **Группа П85**

ГОСУДАРСТВЕННЫЙ СТАНДАРТ СОЮЗА ССР

СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ. ЗАЩИТА КРИПТОГРАФИЧЕСКАЯ. АЛГОРИТМ КРИПТОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ

ГОСТ 28147-89

Срок действия **с 01.07.90 г**.

Настоящий стандарт устанавливает единый алгоритм криптографического преобразования для систем обработки информации в сетях электронных вычислительных машин (ЭВМ), отдельных вычислительных комплексах и ЭВМ, который определяет правила **шифрования** данных и выработки **имитовставки**.

Алгоритм криптографического преобразования предназначен для **аппаратной** или **программной** реализации, удовлетворяет криптографическим требованиям и по своим возможностям не накладывает ограничений на степень секретности защищаемой информации.

Стандарт обязателен для организаций, предприятий и учреждений, применяющих криптографическую защиту данных, хранимых и передаваемых в сетях ЭВМ, в отдельных вычислительных комплексах или в ЭВМ.

Термины, применяемые в настоящем стандарте, и их определения приведены в приложении 1.

1. СТРУКТУРНАЯ СХЕМА АЛГОРИТМА КРИПТОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ

1.1. Структурная схема алгоритма криптографического преобразования (криптосхема) содержит (см. черт. 1):

Криптосхема содержит следующие элементы:

ключевое запоминающее устройство (**КЗУ**) на **256** бит, состоящее из **8**-ми **32**-разрядных накопителей (**X0, X1, X2, X3, X4, X5, X6, X7**);

четыре **32**-разрядные накопителя (**N1, N2, N3, N_4**);

два **32**-разрядных накопителя (**N5, N6**) с записанными в них постоянными заполнениями **C2**, **C1**;

два **32**-разрядных сумматора по модулю 2^{32} **(СМ1, СМ3)**;

32-разрядный сумматор поразрядного суммирования по модулю **2 (СМ2)**;

сумматор по модулю **2** без ограничений на разрядность (**СМ5**);

блок подстановки (**К**);

регистр циклического сдвига на одиннадцать шагов в сторону старшего разряда (**R**).

1.2. Блок подстановки **К** состоит из восьми узлов замены **К1, К2, К3, К4, К5, К6, К7, К8** с памятью по **64** бита каждый. Поступающий на блок подстановки 32-разрядный вектор разбивается на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в 4_разрядный вектор соответствующим узлом замены, представляющим собой таблицу из шестнадцати строк, содержащих по четыре бита заполнения в строке. Входной вектор определяет адрес строки в таблице, заполнение данной строки является выходным вектором. Затем 4-разрядные выходные векторы последовательно объединяются в 32-разрядный вектор.

1.3. При сложении и циклическом сдвиге двоичных векторов старшими разрядами считаются разряды накопителей с большими номерами.

1.4. При записи ключа (**W1, W2, … W256**), **Wq [0,1], q = 1 … 256**, в КЗУ значение **W1** вводится в 1-й разряд накопителя **X0**, …, значение **W32** вводится в 32-й разряд накопителя **X0**, значение **W33** вводится в 1-й разряд накопителя **X1**, значение **W34** вводится во 2-й разряд накопителя **X1**, …, значение **W64** вводится в 32-й разряд накопителя **X1**, значение **W65** вводится в 1-й разряд накопителя **X2** и т.д., значение **W256** вводится в 32-й разряд накопителя **X7**.

1.5. При перезаписи информации содержимое ***p*-**го разряда одного накопителя (сумматора) переписывается в ***p*-**й разряд другого накопителя (сумматора).

1.6. Значения постоянных заполнений **C1, C2** (констант) накопителей **N6**, **N5** приведены в приложении 2.

1.7. Ключи, определяющие заполнение **КЗУ** и таблиц блока подстановки **К** являются секретными элементами и поставляются в установленном порядке.

Заполнение таблиц блока подстановки **К** является долговременным ключевым элементом, общим для сети ЭВМ.

Организация различных видов связи достигается построением соответствующей ключевой системы. При этом может быть использована возможность выработки ключей (заполнений **КЗУ**) в режиме простой замены и зашифрования их в режиме простой замены с обеспечением имитозащиты для передачи по каналам связи или хранения в памяти ЭВМ.

1.8. В криптосхеме предусмотрены четыре вида работы:

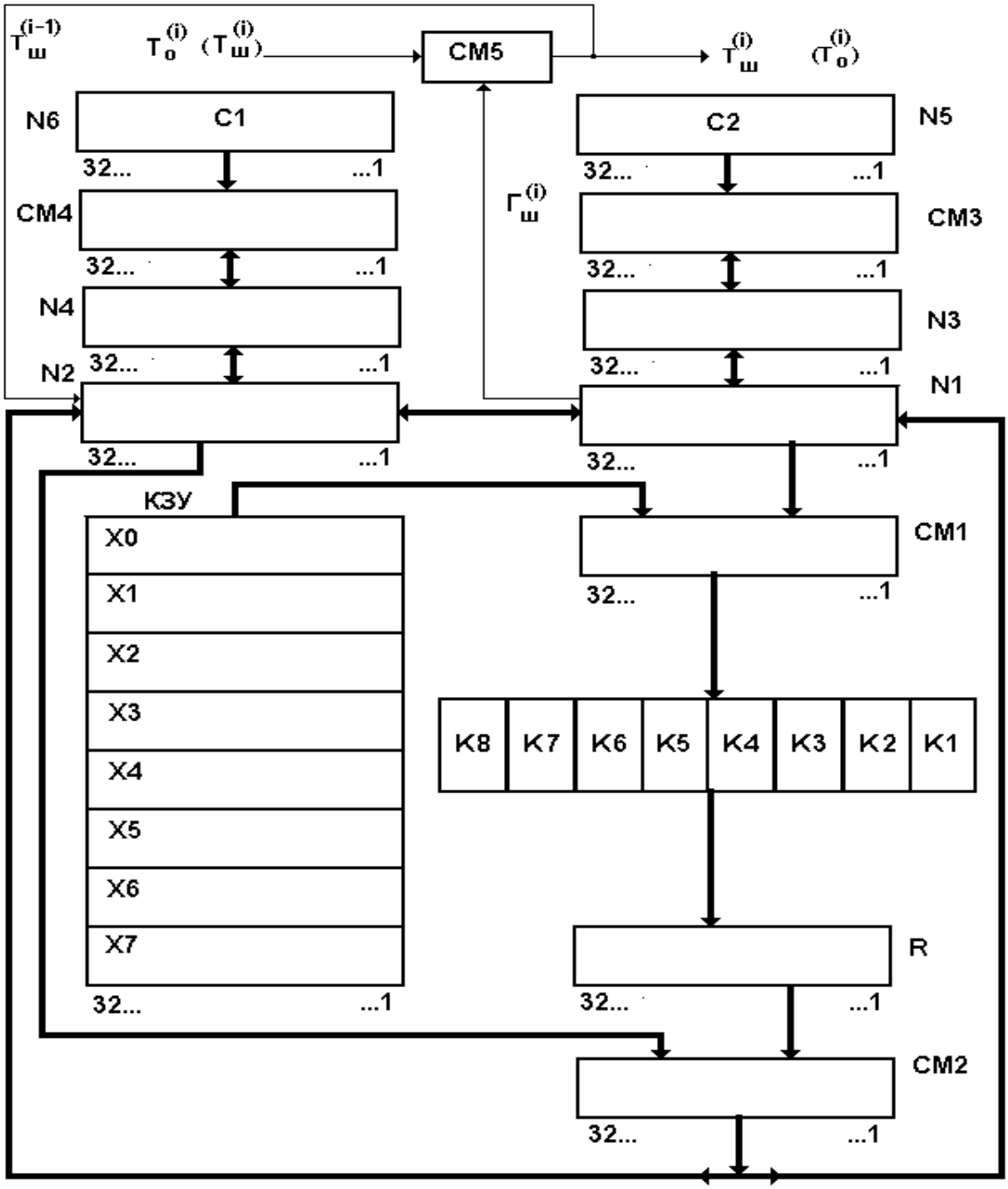
зашифрование (расшифрование) данных в режиме простой замены;

зашифрование (расшифрование) данных в режиме гаммирования;

зашифрование (расшифрование) данных в режиме гаммирования с обратной связью;

режим выработки имитовставки.

Схемы программной реализации алгоритма криптографического преобразования приведены в приложении 3.



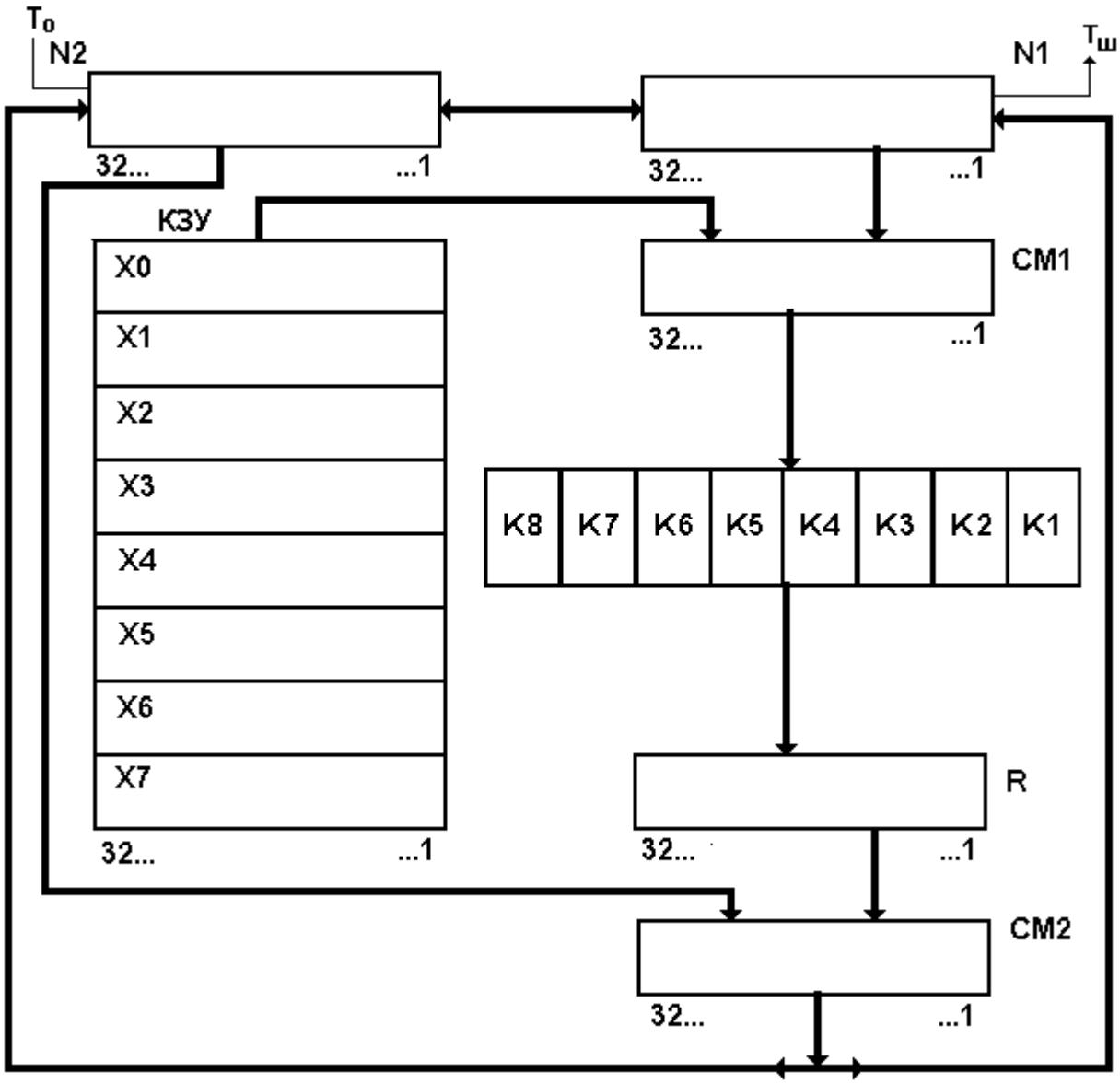
Черт. 1

2. РЕЖИМ ПРОСТОЙ ЗАМЕНЫ

2.1. **Зашифрование открытых данных в режиме простой замены**.

2.1.1. Криптосхема, реализующая алгоритм зашифрования в режиме простой замены, должна иметь вид, указанный на черт. 2.

Открытые данные, подлежащие зашифрованию, разбивают на блоки по 64 бита каждый. Ввод любого блока T_O = **(a1(0), a2(0), ..., a31(0), a32(0), b1(0), b2(0), …, b32(0))** двоичной информации в накопители **N1** и **N2** производятся так, что значение **a1(0)** вводится в 1-й разряд **N1**, значение **a2(0)** вводится во 2-й разряд накопителя и т.д., значение **a32(0)** вводится в 32-й разряд **N1**, значение **b1(0)** вводится в 1-й разряд **N2**, значение **b2(0)** вводится во 2-й разряд накопителя и т.д., значение **b32(0)** вводится в 32-й разряд **N2**. В результате получаем состояние **(a32(0), a31(0), ..., a2(0), a1(0))** накопителя **N1** и состояние **(b32(0), b31(0), ..., b2(0), b1(0))** накопителя **N2**.



Черт. 2

2.1.2. В КЗУ вводится 256 бит ключа. Содержимое восьми 32-разрядных накопителей **X0**, **X1**, **... , X7** имеет вид:

X0 = (W32, W31, ... , W2, W1)

X1 = (W64, W63, ... , W34, W33)

. . .

X7 = (W256, W255, ... , W226, W225)

2.1.3. Алгоритм зашифрования 64-разрядного блока открытых данных в режиме простой замены состоит из 32-х циклов.

В первом цикле начальное заполнение накопителя **N1** суммируется по модулю 2^{32} в сумматоре **СМ1** с заполнением накопителя **X0**, при этом заполнение накопителя **N1** сохраняется.

Результат суммирования преобразуется в блоке подстановке **K** и полученный вектор поступает на вход регистра **R**, где циклически сдвигается на одиннадцать шагов в сторону старших разрядов. Результат сдвига суммируется поразрядно по модулю 2 в сумматоре **СМ2** с 32-разрядным заполнением накопителя **N2**. Полученный в **СМ2** результат записывается в **N1**, при этом старое заполнение **N1** переписывается в **N2**. Первый цикл заканчивается.

Последующие циклы осуществляются аналогично, при этом во 2-м цикле из **КЗУ** считывается заполнение **X1**, в 3-ем цикле из **КЗУ** считывается заполнение **X2** и т.д., в 8-ом цикле из **КЗУ** считывается заполнение **X7**. В циклах с 9-го по 16-ый, а также в циклах с 17-го по 24-ый заполнения из **КЗУ** считываются в том же порядке:

X0, **X1**, **X2**, **X3**, **X4**, **X5, X6, X7.**

В последних восьми циклах с 25-ый по 32-ой порядок считывания заполнений **КЗУ** обратный:

X7, **X6**, **X5**, **X4**, **X3**, **X2, X1, X0.**

Таким образом, при зашифровании в 32 циклах осуществляется следующий порядок выбора заполнений накопителей:

X0, **X1**, **X2**, **X3**, **X4**, **X5, X6, X7,** **X0**, **X1**, **X2**, **X3**, **X4**, **X5, X6, X7,**

X0, **X1**, **X2**, **X3**, **X4**, **X5, X6, X7,** **X7**, **X6**, **X5**, **X4**, **X3**, **X2, X1, X0.**

В 32-ом цикле результат из сумматора **СМ2** вводится в накопитель **N2**, а в накопителе **N1** сохраняется старое заполнение.

Полученные после 32-го цикла зашифрования заполнения накопителей **N1** и **N2** являются блоком зашифрованных данных, соответствующим блоку открытых данных.

2.1.4. Уравнения зашифрования в режиме простой замены имеют вид:

$$\begin{cases} a(j) = (a(j-1) \boxplus X_{(j-1)\,(\mathrm{mod}\,8)}) KR \oplus b(j-1); \\ b(j) = a(j-1); \end{cases}$$

при $j = 1 \ldots 24;$

$$\begin{cases} a(j) = (a(j-1) \boxplus X_{(32-j)\,(\mathrm{mod}\,8)}) KR \oplus b(j-1); \\ b(j) = a(j-1); \end{cases}$$

при $j = 25 \ldots 31;$

$$\begin{cases} a(32) = a(31) \\ b(32) = (a(31) \boxplus X_0)\, KR \oplus b(31); \end{cases}$$

при $j = 32,$

где **a(0) = (a32(0), a31(0), ..., a2(0), a1(0))** – начальное заполнение накопителя **N1** перед первым циклом зашифрования;

b(0) = (b32(0), b31(0), ..., b2(0), b1(0)) – начальное заполнение накопителя **N2** перед первым циклом зашифрования;

a(j) = (a32(j), a31(j), ..., a2(j), a1(j)) – заполнение **N1** после j-го цикла зашифрования, **j = 1 … 32;**

b(j) = (b32(j), b31(j), ..., b2(j), b1(j)) – заполнение **N2** после j-го цикла зашифрования, **j = 1 … 32;**

Знак $\oplus$ означает поразрядное суммирование 32-разрядных векторов по модулю **2**;

Знак $\boxplus$ означает суммирование 32-разрядных векторов по модулю 2^{32}. Правила суммирования по модулю 2^{32} приведены в приложении 4;

R – операция циклического сдвига на одиннадцать шагов в сторону старших разрядов, т. е.

$$R(r_{32}, r_{31}, \ldots, r_{22}, r_{21}, r_{20}, \ldots, r_2, r_1) = (r_{21}, r_{20}, \ldots, r_2, r_1, r_{32}, r_{31}, \ldots, r_{22}).$$

2.1.5. 64-разрядный блок зашифрованных данных $T_{Ш}$ выводится из накопителей **N1**, **N2** в следующем порядке: из 1-го, 2-го,..., 32-го разряда накопителя **N1**, затем из 1-го, 2-го, ..., 32-го разрядов накопителя **N2**, т. е.

$$T_{Ш} = (a_1(32), a_2(32), \ldots, a_{32}(32), b_1(32), b_2(32), \ldots, b_{32}(32)).$$

Остальные блоки открытых данных в режиме простой замены зашифровываются аналогично.

2.2. Расшифрование зашифрованных данных в режиме простой замены

2.2.1. Криптосхема, реализующая алгоритм расшифрования в режиме простой замены, имеет тот же вид (см. черт. 2), что и при зашифровании. В КЗУ вводятся 256 бит того же ключа, на котором осуществлялось зашифрование открытых данных. Зашифрованные данные, подлежащие расшифрованию, разбиты на блоки по 64 бита в каждом. Ввод любого блока

$$T_{Ш} = (a_1(32), a_2(32), \ldots, a_{32}(32), b_1(32), b_2(32), \ldots, b_{32}(32))$$

в накопители **N1** и **N2** производятся так, что значение $a_1(32)$ вводится в 1-й разряд **N1**, значение $a_2(32)$ вводится во 2-й разряд **N1** и т.д., значение $a_{32}(32)$ вводится в 32-й разряд **N1**; значение $b_1(32)$ вводится в 1-й разряд **N2** и т.д., значение $b_{32}(32)$ вводится в 32-й разряд **N2**.

2.2.2. Расшифрование осуществляется по тому же алгоритму, что и зашифрование открытых данных, с тем изменением, что заполнения накопителей **X0, X1, ..., X7** считываются из КЗУ в циклах расшифрования в следующем порядке:

X0, X1, X2, X3, X4, X5, X6, X7, X7, X6, X5, X4, X3, X2,. X1, X0
X7, X6, X5, X4, X3, X2,. X1, X0, X7, X6, X5, X4, X3, X2,. X1, X0.

2.2.3. Уравнения расшифрования имеют вид:

$$\begin{cases} a(32-j) = (a(32-j+1) \boxplus X_{(j-1)}) KR \oplus b(32-j+1); \\ b(32-j) = a(32-j+1); \end{cases}$$

при **j = 1 … 8;**

$$\begin{cases} a(32-j) = (a(32-j+1) \boxplus X_{(32-j)\,(mod\ 8)}) KR \oplus b(32-j+1); \\ b(32-j) = a(32-j+1); \end{cases}$$

при **j = 9 … 31;**

$$\begin{cases} a(0) = a(1) \\ b(0) = (a(1) \boxplus X_0)\, KR \oplus b(1); \end{cases}$$

при **j = 32,**

2.2.4. Полученные после 32-х циклов работы заполнения накопителей **N1** и **N2** составляют блок открытых данных

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

соответствующий блоку зашифрованных данных, при этом значение $a_1(0)$ блока T_0 соответствует содержимому 1-го разряда **N1**, значение $a_2(0)$ соответствует содержимому 2-го разряда **N1** и т.д., значение $a_{32}(0)$ соответствует содержимому 32-го разряда **N1**, значение $b_1(0)$ соответствует содержимому 1-го разряда **N2**, значение $b_2(0)$ соответствует содержимому 2-го разряда **N2** и т.д., значение $b_{32}(0)$ соответствует содержимому 32-го разряда **N2**.

Аналогично расшифровываются остальные блоки зашифрованных данных.

2.3. Алгоритм зашифрования в режиме простой замены 64-битового блока To обозначается через /\, т.е.:

$$/\backslash\ (T_O) = /\backslash\ (a(0), b(0)) = (a(32), b(32)) = T_Ш.$$

2.4. Режим простой замены допускается использовать для зашифрования (расшифрования) данных только в случаях, приведенных в п. 1.7.

3. РЕЖИМ ГАММИРОВАНИЯ

3.1. Зашифрование открытых данных в режиме гаммирования

3.1.1. Криптосхема, реализующая алгоритм зашифрования данных в режиме гаммирования, имеет вид, указанный на черт. 3.

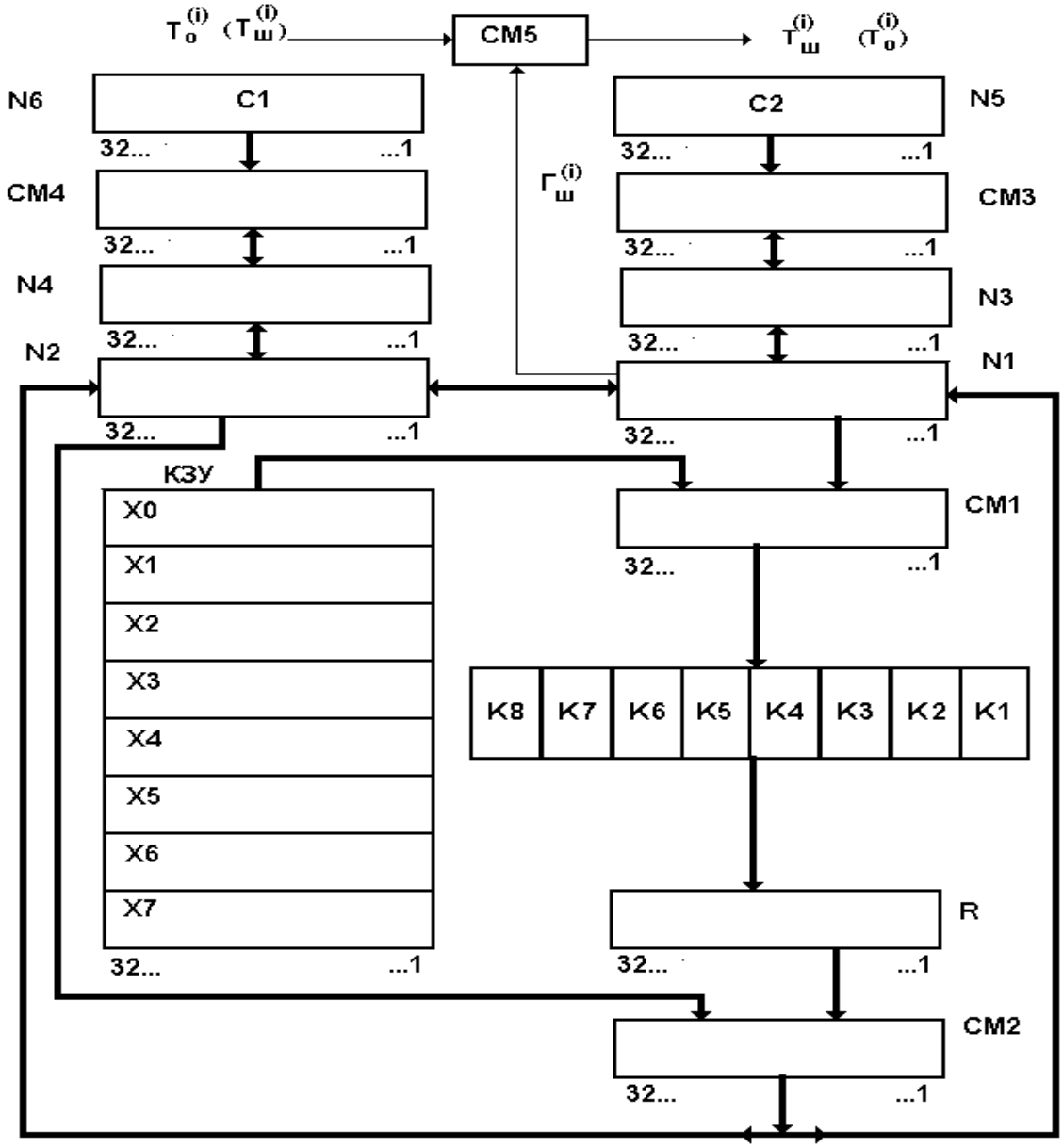
Открытые данные, разбитые на **64**-разрядные блоки $To^{(1)}$, $To^{(2)}$, ..., $To^{(M-1)}$, $To^{(M)}$, зашифровываются в режиме гаммирования путем поразрядного суммирования по модулю **2** в сумматоре **СМ5** с гаммой шифра $Г_Ш$, которая вырабатывается блоками по **64** бита, т. е.

$$Г_Ш = (Г_Ш^{(1)}, Г_Ш^{(2)}, ..., Г_Ш^{(M-1)}, Г_Ш^{(M)})$$

где **M** - определяется объемом шифруемых данных.

$Г_Ш^{(i)}$ - i-й 64-разрядный блок, i = 1 … M, число двоичных разрядов в блоке $To^{(M)}$ может быть меньше 64, при этом неиспользованная для зашифрования часть гаммы шифра из блока $Г_Ш^{(M)}$ отбрасывается.

3.1.2. В **КЗУ** вводятся **256** бит ключа. **В** накопителе N_1, N_2 вводится **64**-разрядная двоичная последовательность (синхропосылка) $S = (S_1, S_2, ..., S_{64})$, являющаяся исходным заполнением этих накопителей для последующей выработки **M** блоков гаммы шифра. Синхропосылка вводится в N_1 и N_2 так, что значение S_1 вводится в 1-й разряд N_1, значение S_2 вводится во 2-й разряд N_1 и т.д., значение S_{32} вводится в 32-й разряд N_1, значение S_{33} вводится в 1-й разряд N_2, значение S_{34} вводится во 2-й разряд N_2 и т.д., значение S_{64} вводится в 32-й разряд N_2.



Черт. 3

3.1.3. Исходное заполнение накопителей **N1** и **N2** (синхропосылка **S)** зашифровывается в режиме простой замены в соответствии с требованиями п. 2.1. Результат зашифрования **/\(S) = (Y_0, Z_0)** переписывается в **32**-разрядные накопители **N3** и **N4** так, что заполнение **N1** переписывается в **N3**, а заполнение **N2** переписывается в **N4**.

3.1.4. Заполнение накопителя **N4** суммируется по модулю **(2^{32}-1**) в сумматоре **CM4** с **32**-разрядной константой **C1** из накопителя **N6**, результат записывается в **N4**. Заполнение накопителя **N3** суммируется по модулю 2^{32} в сумматоре **CM3** с **32**-разрядной константой **C2** из накопителя **N5**, результат записывается в **N3**.

Заполнение **N3** переписывается в **N1**, а заполнение **N4** переписывается в **N2**, при этом заполнение **N3**, **N4** сохраняется.

Заполнение **N1** и **N2** зашифровывается в режиме простой замены. Полученное в результате зашифрования заполнение **N1**, **N2** образует первый **64**-разрядный блок гаммы шифра $\Gamma_{Ш}^{(1)}$), который суммируется поразрядно по модулю **2** в сумматоре **CM5** с первым **64**-разрядным блоком открытых данных

$$T_O^{(1)} = (t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)}).$$

В результате суммирования получается **64**-разрядный блок зашифрованных данных

$$\mathbf{T_Ш^{(1)} = (T_1^{(1)}, T_2^{(1)}, \ldots, T_{63}^{(1)}, T_{64}^{(1)})}.$$

Значение $\mathbf{T_1^{(1)}}$ блока $\mathbf{T_Ш^{(1)}}$ является результатом суммирования по модулю **2** в **СМ5** значения $\mathbf{t_1^{(1)}}$ из блока $\mathbf{T_О^{(1)}}$ со значением 1-го разряда **N1**, значение $\mathbf{T_2^{(1)}}$ блока $\mathbf{T_Ш^{(1)}}$ является результатом суммирования по модулю **2** в **СМ5** значения $\mathbf{t_2^{(1)}}$ из блока $\mathbf{T_О^{(1)}}$ со значением 2-го разряда **N1** и т.д., $\mathbf{T_{64}^{(1)}}$ блока $\mathbf{T_Ш^{(1)}}$ является результатом суммирования по модулю **2** в **СМ5** значения $\mathbf{t_{64}^{(1)}}$ из блока $\mathbf{T_О^{(1)}}$ со значением 32-го разряда **N2**,

3.1.5. Для получения следующего 64-разрядного блока гаммы шифра $\mathbf{Г_Ш^{(2)}}$ заполнение **N4** суммируется по модулю $\mathbf{(2^{32}-1)}$ в сумматоре **СМ4** с константой **С1** из **N6**, заполнение **N3** суммируется по модулю $\mathbf{2^{32}}$ в сумматоре **СМ3** с константой **С2** из **N5**. Новое заполнение **N3** переписывается в **N1**, а новое заполнение **N4** переписывается в **N2**, при этом заполнение **N3** и **N4** сохраняется.

Заполнение **N1** и **N2** зашифровывается в режиме простой замены в соответствии с требованиями п. 2.1. Полученное в результате зашифрования заполнение **N1**, **N2** образует второй 64-разрядный блок гаммы шифра $\mathbf{Г_Ш^{(2)}}$, который суммируется поразрядно по модулю **2** в сумматоре **СМ5** со вторым блоком открытых данных $\mathbf{T_О^{(2)}}$. Аналогично вырабатываются блоки гаммы шифра $\mathbf{Г_Ш^{(3)}, Г_Ш^{(4)}, ..., Г_Ш^{(M)}}$ и зашифровываются блоки открытых данных $\mathbf{T_О^{(3)}, T_О^{(4)}, ..., T_О^{(M)}}$. Если длина последнего М-го блока открытых данных $\mathbf{T_О^{(M)}}$ меньше 64 бит, то из последнего М-го блока гаммы шифра $\mathbf{Г_Ш^{(M)}}$ для зашифровывания используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

3.1.6. В канал связи или память ЭВМ передаются синхропосылка **S** и блоки зашифрованных данных $\mathbf{T_Ш^{(1)}, T_Ш^{(2)}, ..., T_Ш^{(M-1)}, T_Ш^{(M)}}$.

3.1.7. Уравнение зашифрования имеет вид

$$T_Ш^{(i)} = /\backslash(Y_{i-1} \boxplus C2, Z_{i-1} \boxplus` C1) \oplus T_О^{(i)} = Г_Ш^{(i)} \oplus T_О^{(i)},$$

$i = 1 \ldots M,$

где $\boxplus`$ - означает суммирование 32-разрядных заполнений по модулю $\mathbf{(2^{32} – 1)}$;

$\boxplus$ - означает суммирование 32-разрядных заполнений по модулю $\mathbf{2^{32}}$;

$\oplus$ - означает поразрядное суммирование по модулю **2** двух заполнений;

$\mathbf{Y_i}$ - содержимое накопителя **N3** после зашифрования i-го блока открытых данных $\mathbf{T_О^{(i)}}$;

$\mathbf{Z_i}$ - содержимое накопителя **N4** после зашифрования i-го блока открытых данных $\mathbf{T_О^{(i)}}$;

$$\mathbf{(Y_0, Z_0) = /\backslash(S).}$$

3.2. Расшифрование зашифрованных данных в режиме гаммирования

3.2.1. При расшифровании криптосхема имеет тот же вид, что и при зашифровании (см. черт. 3). В **КЗУ** вводятся **256** бит ключа, с помощью которого осуществлялось зашифрование данных $\mathbf{T_О^{(1)}, T_О^{(2)}, \ldots, T_О^{(M)}}$. Синхропосылка **S** вводится в накопители **N1** и **N2** и аналогично пп. 3.1.2 - 3.1.5 осуществляется процесс выработки **M** блоков гаммы шифра $\mathbf{Г_Ш^{(1)}, Г_Ш^{(2)}, ..., Г_Ш^{(M)}}$. Блоки зашифрованных данных $\mathbf{T_Ш^{(1)}, T_Ш^{(2)}, \ldots, T_Ш^{(M)}}$ суммируются поразрядно по модулю **2** в сумматоре **СМ5** с блоками гаммы шифра, в результате получаются блоки открытых данных $\mathbf{T_О^{(1)}, T_О^{(2)}, \ldots, T_О^{(M)}}$, при этом $\mathbf{T_О^{(M)}}$ может содержать меньше 64 разрядов.

3.2.2. Уравнение расшифрования имеет вид

$$T_О^{(i)} = /\backslash(Y_{i-1} \boxplus C2, Z_{i-1} \boxplus` C1) \oplus T_О^{(i)} = Г_Ш^{(i)} \oplus T_О^{(i)},$$

$i = 1 \ldots M.$

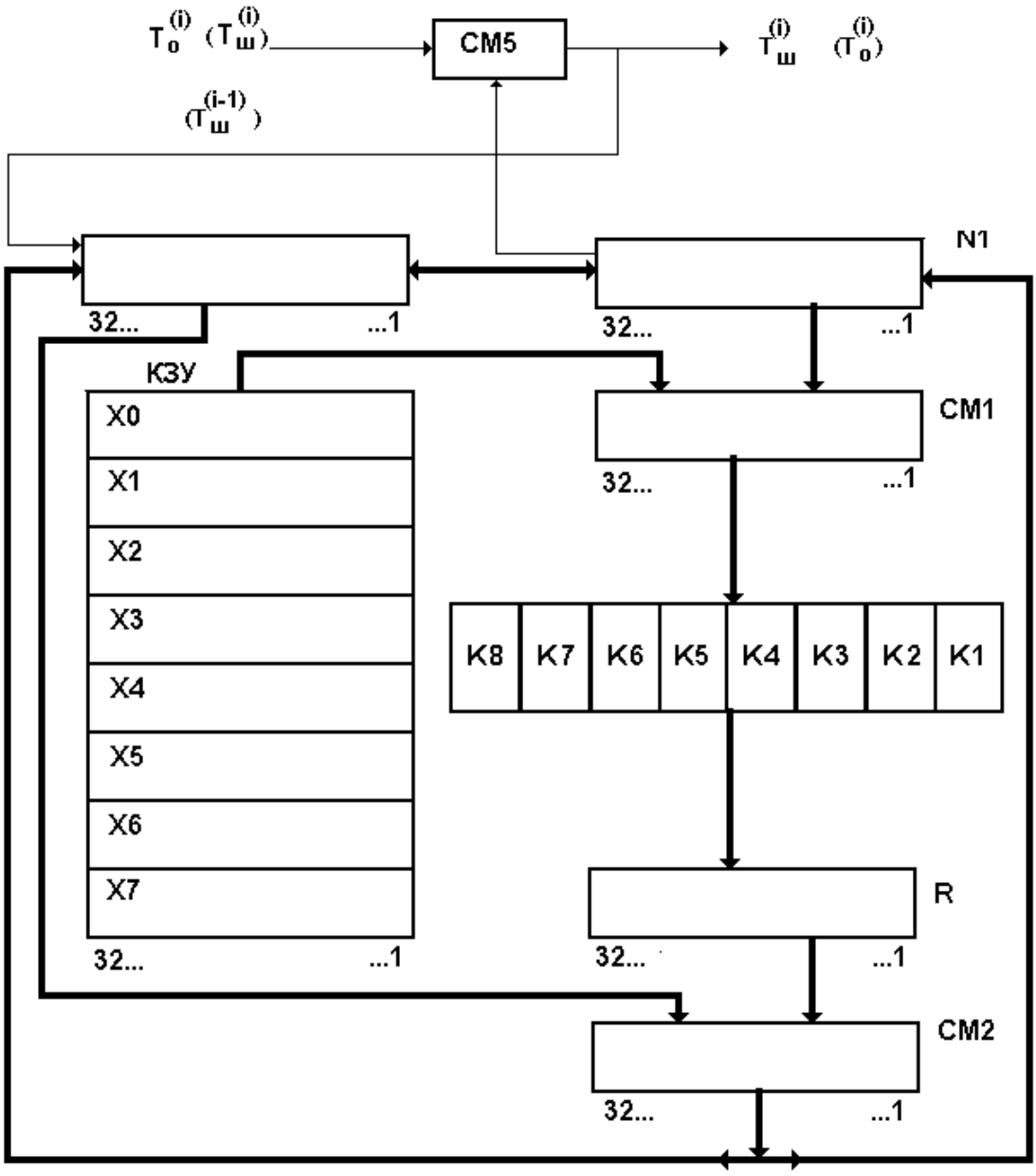
4. РЕЖИМ ГАММИРОВАНИЯ С ОБРАТНОЙ СВЯЗЬЮ

4.1. Зашифрование открытых данных в режиме гаммирования с обратной связью

4.1.1. Криптосхема, реализующая алгоритм зашифрования в режиме гаммирования с обратной связью, имеет вид, указанный на черт. 4.

Открытые данные, разбитые на **64**-разрядные блоки $\mathbf{T_О^{(1)}, T_О^{(2)}, \ldots, T_О^{(M)}}$, зашифровываются в режиме гаммирования с обратной связью путем поразрядного суммирования по модулю **2** в сумматоре **СМ5** с гаммой шифра $\mathbf{Г_Ш}$, которая вырабатывается блоками по **64** бита, т. е. $\mathbf{Г_Ш = (Г_Ш^{(1)}, Г_Ш^{(2)}, ..., Г_Ш^{(M-1)}, Г_Ш^{(M)})}$, где М определяется объемом

открытых данных, $\Gamma_{Ш}^{(1)}$ – i – й 64-разрядный блок, **i = 1 … M**. Число двоичных разрядов в блоке $T_O^{(M)}$ может быть меньше 64.



Черт. 4

4.1.2. В **КЗУ** вводится **256** бит ключа. Синхропосылка **S = (S_1, S_2, …, S_{64})** из **64** бит вводится в **N1** и **N2** аналогично п. 3.1.2.

4.1.3. Исходное заполнение **N1** и **N2** зашифровывается в режиме простой замены в соответствии с требованиями п. 2.1. Полученное в результате зашифрования заполнение **N1** и **N2** образует первый **64**-разрядный блок гаммы шифра $\Gamma_{Ш}^{(1)} = /\backslash(S)$, который суммируется поразрядно по модулю **2** в сумматоре **СМ5** с первым **64**-разрядным блоком открытых данных $T_O^{(1)} = (t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)})$.

В результате получается **64**-разрядный блок зашифрованных данных

$$T_Ш^{(1)} = (T_1^{(1)}, T_2^{(1)}, \ldots, T_{63}^{(1)}, T_{64}^{(1)}).$$

4.1.4. Блок зашифрованных данных $T_Ш^{(1)}$ одновременно является также исходным состоянием **N1**, **N2** для выработки второго блока гаммы шифра $\Gamma_{Ш}^{(2)}$ и по обратной связи записывается в указанные накопители. При этом значение $T_1^{(1)}$ вводится в 1-й разряд **N1**, значение $T_2^{(1)}$ вводится во 2-й разряд **N1** и т.д., значение $T_{32}^{(1)}$ вводится в 32-й разряд **N1**,

значение $\mathbf{T_{33}^{(1)}}$ вводится в 1-й разряд **N2**, значение $\mathbf{T_{34}^{(1)}}$ вводится во 2-й разряд **N2** и т.д., значение $\mathbf{T_{64}^{(1)}}$ вводится в 32-й разряд **N2**.

Заполнение **N1, N2** зашифровывается в режиме простой замены в соответствии с требованиями п. 2.1. Полученное в результате зашифрования заполнение **N1, N2** образует второй **64**-разрядный блок гаммы шифра $\mathbf{Г_Ш^{(2)}}$, который суммируется поразрядно по модулю **2** в сумматоре **СМ5** со вторым блоком открытых данных $\mathbf{T_O^{(2)}}$.

Выработка последующих блоков гаммы шифра $\mathbf{Г_Ш^{(i)}}$ и зашифрование соответствующих блоков открытых данных $\mathbf{T_O^{(i)}}$.**(i = 3 … M)** производится аналогично. Если длина последнего М-го блока открытых данных $\mathbf{T_O^{(M)}}$ меньше 64 разрядов, то из $\mathbf{Г_Ш^{(M)}}$ используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

4.1.5. Уравнение зашифровывания в режиме гаммирования с обратной связью имеют вид:

$$\begin{cases} \mathbf{T_Ш^{(1)} = Λ(S) \oplus T_O^{(1)} = Г_Ш^{(1)} \oplus T_O^{(1)};} \\ \mathbf{T_Ш^{(i)} = Λ(T_Ш^{(I-1)}) \oplus T_O^{(i)} = Г_Ш^{(i)} \oplus T_O^{(i)}, i = 2 \ldots M.} \end{cases}$$

4.1.6. В канал связи или память ЭВМ передаются синхропосылка **S** и блоки зашифрованных данных $\mathbf{T_Ш^{(1)}, T_Ш^{(2)}, ..., T_Ш^{(M)}}$.

4.2. Расшифрование зашифрованных данных в режиме гаммирования с обратной связью

4.2.1. При расшифровании криптосхема имеет тот же вид (см. черт. 4), что и при зашифровании.

В **КЗУ** вводятся **256** бит того же ключа, на котором осуществлялось зашифрование $\mathbf{To^{(1)}, To^{(2)},..., To^{(M)}}$. Синхропосылка **S** вводится в **N1, N2** аналогично п. 3.1.2.

4.2.2. Исходное заполнение **N1, N2** (синхропосылка **S**) зашифровывается в режиме простой замены согласно подразделу 2.1. Полученное в результате зашифрования заполнение **N1,** $\mathbf{N_2}$ образует первый блок гаммы шифра $\mathbf{Г_Ш^{(1)} = Λ(S)}$, который суммируется поразрядно по модулю **2** в сумматоре **СМ5** с блоком зашифрованных данных $\mathbf{T_Ш^{(1)}}$. В результате получается первый блок открытых данных $\mathbf{T_O^{(1)}}$.

4.2.3. Блок зашифрованных данных $\mathbf{T_Ш^{(1)}}$ является исходным заполнением **N1, N2** для выработки второго блока гаммы шифра $\mathbf{Г_Ш^{(2)}}$. Блок $\mathbf{T_Ш^{(1)}}$ записывается в **N1, N2** в соответствии с требованиями п. 4.1.4. Полученное заполнение **N1, N2** зашифровывается в режиме простой замены в соответствии с требованиями п. 2.1, полученный в результате блок $\mathbf{Г_Ш^{(2)}}$ суммируется поразрядно по модулю **2** в сумматоре **СМ5** со вторым блоком зашифрованных данных $\mathbf{T_Ш^{(2)}}$. В результате получается блок открытых данных $\mathbf{T_O^{(2)}}$.

Аналогично в **N1,** $\mathbf{N_2}$ последовательно записываются блоки из зашифрованных данных $\mathbf{T_Ш^{(2)}, T_Ш^{(3)}, ..., T_Ш^{(M-1)}}$, из которых в режиме простой замены вырабатываются блоки гаммы шифра $\mathbf{Г_Ш^{(3)}, Г_Ш^{(4)}, ..., Г_Ш^{(M)}}$. Блоки гамма шифра суммируются поразрядно по модулю **2** в сумматоре **СМ5** с блоками зашифрованных данных $\mathbf{T_Ш^{(3)}, T_Ш^{(4)}, ..., T_Ш^{(M)}}$, в результате получаются блоки открытых данных $\mathbf{T_O^{(3)}, T_O^{(4)}, …, T_O^{(M)}}$, при этом длина последнего блока открытых данных $\mathbf{T_O^{(M)}}$ может содержать меньше 64 разрядов.

4.2.4. Уравнение расшифрования в режиме гаммирования с обратной связью имеет вид:

$$\begin{cases} \mathbf{T_O^{(1)} = Λ(S) \oplus T_Ш^{(1)} = Г_Ш^{(1)} \oplus T_Ш^{(1)},} \\ \mathbf{T_O^{(i)} = Λ(T^{(i-1)}) \oplus T_Ш^{(i)} = Г_Ш^{(i)} \oplus T_Ш^{(i)},} \text{ i = 2 M.} \end{cases}$$

5. РЕЖИМ ВЫРАБОТКИ ИМИТОВСТАВКИ

5.1. Для обеспечения имтозащиты открытых данных, состоящих из **M 64**-разрядных блоков $\mathbf{To^{(1)}, To^{(2)}, ..., To^{(M)}}$ где **M 2**, вырабатывается дополнительный блок из ***l*** бит (имитовставка $\mathbf{И_l}$). Процесс выработки имитовставки единообразен для всех режимов шифрования.

5.2. Первый блок открытых данных

$$\mathbf{T_O^{(1)} = (t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)}) = (a_1^{(1)}(0), a_2^{(1)}(0), \ldots, a_{32}^{(1)}(0), b_1^{(1)}(0), b_2^{(1)}(0), \ldots, b_{32}^{(1)}(0))}$$

записывается в накопители **N1, N2**, при этом значение $\mathbf{t_1^{(1)} = a_1^{(1)}(0)}$ вводится в 1-й разряд **N1**, значение $\mathbf{t_2^{(1)} = a_2^{(1)}(0)}$ вводится во 2-й разряд **N1** и т.д., значение $\mathbf{t_{32}^{(1)} = a_1^{(1)}(0)}$ вводится в 32-й разряд **N1**, значение $\mathbf{t_{33}^{(1)} = b_1^{(1)}(0)}$ вводится в 1-й разряд **N2** и т.д., значение $\mathbf{t_{64}^{(1)} = b_{32}^{(1)}(0)}$ вводится в 32-й разряд **N2**.

5.3. Заполнение **N1** и **N2** подвергается преобразованию, соответствующему первым **16** циклам алгоритма зашифрования в режиме простой замены (см. подраздел 2.1). В **КЗУ** при этом находится тот же ключ, которым зашифровываются блоки открытых данных $\mathbf{To^{(1)}, To^{(2)}, \ldots, To^{(M)}}$ в соответствующие блоки зашифрованных данных $\mathbf{T_Ш^{(1)}, T_Ш^{(2)}, \ldots, T_Ш^{(M)}}$.

Полученное после **16** циклов работы заполнение **N1, N2**, имеющее вид

$$\mathbf{(a_1^{(1)}(16), a_2^{(1)}(16), \ldots, a_{32}^{(1)}(16), b_1^{(1)}(16), b_2^{(1)}(16), \ldots, b_{32}^{(1)}(16))}$$

суммируется в **СМ5** по модулю **2** со вторым блоком открытых данных

$$\mathbf{T_O^{(2)} = (t_1^{(2)}, t_2^{(2)}, \ldots, t_{63}^{(2)}, t_{64}^{(2)})}.$$

Результат суммирования

$$\mathbf{(a_1^{(1)}(16)\ \ t_1^{(2)}, a_2^{(1)}(16)\ \ t_2^{(2)}, \ldots, a_{32}^{(1)}(16)\ \ t_{32}^{(2)}, b_1^{(1)}(16)\ \ t_{33}^{(2)}, \ldots, b_{32}^{(1)}(16)\ \ t_{64}^{(2)}) =}$$
$$\mathbf{= (a_1^{(2)}(0), a_2^{(2)}(0), \ldots, a_{32}^{(2)}(0), b_1^{(2)}(0), b_2^{(2)}(0), \ldots, b_{32}^{(2)}(0))}$$

заносится в $\mathbf{N_1, N_2}$ и подвергается преобразованию, соответствующему первым **16** циклам алгоритма зашифрования в режиме простой замены.

Полученное заполнение **N1, N2** суммируется в **СМ5** по модулю 2 с третьим блоком открытых данных $\mathbf{T_O^{(2)}}$ и т. д., последний блок $\mathbf{T_O^{(M)}}$, при необходимости дополненный до полного 64-разрядного блока нулями, суммируется в **СМ5** по модулю 2 с заполнением **N1, N2**

$$\mathbf{(a_1^{(M-1)}(16), a_2^{(M-1)}(16), \ldots, a_{32}^{(M-1)}(16), b_1^{(M-1)}(16), b_2^{(M-1)}(16), \ldots, b_{32}^{(M-1)}(16)).}$$

Результат суммирования

$$\mathbf{(a_1^{(M-1)}(16)\ \ t_1^{(M)}, a_2^{(M-1)}(16)\ \ t_2^{(M)}, \ldots, a_{32}^{(M-1)}(16)\ \ t_{32}^{(M)}, b_1^{(M-1)}(16)\ \ t_{33}^{(M)}, \ldots, b_{32}^{(M-1)}(16)\ \ t_{64}^{(M)}) =}$$
$$\mathbf{= (a_1^{(M)}(0), a_2^{(M)}(0), \ldots, a_{32}^{(M)}(0), b_1^{(M)}(0), b_2^{(M)}(0), \ldots, b_{32}^{(M)}(0))}$$

заносится в $\mathbf{N_1, N_2}$ и зашифровывается в режиме простой замены по первым **16** циклам работы алгоритма. Из полученного заполнения накопителей $\mathbf{N_1}$ **и** $\mathbf{N_2}$

$$\mathbf{(a_1^{(M)}(16), a_2^{(M)}(16), \ldots, a_{32}^{(M)}(16), b_1^{(M)}(16), b_2^{(M)}(16), \ldots, b_{32}^{(M)}(16))}$$

выбирается отрезок $\mathbf{И}_l$ (имитовставка) длиной $\boldsymbol{l}$ бит:

$$\mathbf{И}_l = \mathbf{[a_{32-l+1}^{(M)}(16), a_{32-l+2}^{(M)}(16), \ldots, a_{32}^{(M)}(16)].}$$

Имитовставка $\mathbf{И}_l$ передается по каналу связи или в память ЭВМ в конце зашифрованных данных, т. е. $\mathbf{T_Ш^{(1)}, T_Ш^{(2)}, \ldots, T_Ш^{(M)}, И}_l$.

5.4. Поступившие зашифрованные данные $\mathbf{T_Ш^{(1)}, T_Ш^{(2)}, \ldots, T_Ш^{(M)}}$ расшифровываются, из полученных блоков открытых данных $\mathbf{To^{(1)}, To^{(2)}, \ldots, To^{(M)}}$ аналогично п. 5.3 вырабатывается имитовставка $\mathbf{И}_l$**'**, которая затем сравнивается с имитовставкой $\mathbf{И}_l$, полученной вместе с зашифрованными данными из канала связи или из памяти ЭВМ. В случае несовпадения имитовставок полученные блоки открытых данных $\mathbf{To^{(1)}, To^{(2)}, \ldots, To^{(M)}}$ считаются ложными.

Выработка имитовставки $\mathbf{И}_l$ **(**$\mathbf{И}_l$**')** может производится или перед зашифрованием (после расшифрования) всего сообщения, или параллельно с зашифрованием (расшифрованием) по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (адресную часть, отметку времени, синхропосылку и т.д.) и не зашифровываться.

Значение параметра $\boldsymbol{l}$ (длина двоичных разрядов в имитовставке) определяется действующими криптографическими требованиями, при этом учитывается, что вероятность навязывания ложных данных равна $\mathbf{2^{-l}}$.

ПРИЛОЖЕНИЕ 1

(Справочное)

ТЕРМИНЫ, ПРИМЕНЯЕМЫЕ В НАСТОЯЩЕМ СТАНДАРТЕ, И ИХ ОПРЕДЕЛЕНИЯ

Термин	Определение
Алгоритм	По ГОСТ 19781-90
Гаммирование	Процесс наложения по определенному закону гаммы шифра на открытые данные
Гамма шифра	Псевдослучайная двоичная последовательность, вырабатываемая по заданному алгоритму для зашифрования открытых данных и расшифрования зашифрованных данных
Данные	По ГОСТ 15971-90
Зашифрование данных	Процесс преобразования открытых данных в зашифрованные при помощи шифра
Имитозащита	Защита системы шифрованной связи от навязывания ложных данных
Имитовставка	Отрезок информации фиксированной длины, полученной по определенному правилу из открытых данных и ключа и добавленной к зашифрованным данным для обеспечения имитозащиты
Канал связи	По ГОСТ 17657-79
Ключ	Конкретное некоторое состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор одного преобразования из совокупности всевозможных для данного алгоритма преобразований
Криптографическая защита	Защита данных при помощи криптографического преобразования данных
Криптографическое преобразование	Преобразование данных при помощи шифрования и (или) выработки имитовставки
Расшифрование данных	Процесс преобразования зашифрованных данных в открытые при помощи шифра
Синхропосылка	Значения исходных открытых параметров алгоритма криптографического преобразования
Уравнение зашифрования	Соотношение, выражающее процесс образования зашифрованных данных из открытых данных в результате преобразований, заданных алгоритмом криптографического преобразования
Уравнение расшифрования	Соотношение, выражающее процесс образования открытых данных из зашифрованных данных в результате преобразований, заданных алгоритмом криптографического преобразования
Шифр	Совокупность обратимых преобразований множества возможных открытых данных на множество возможных зашифрованных данных, осуществляемых по определенным правилам с использованием ключей
Шифрование	Процесс зашифрования или расшифрования

ПРИЛОЖЕНИЕ 2

(Обязательное)

ЗНАЧЕНИЯ КОНСТАНТ С!, С2

1. Константа С1 имеет вид

Разряд накопителя N6	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
Значение разряда	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Разряд накопителя N6	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Значение разряда	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

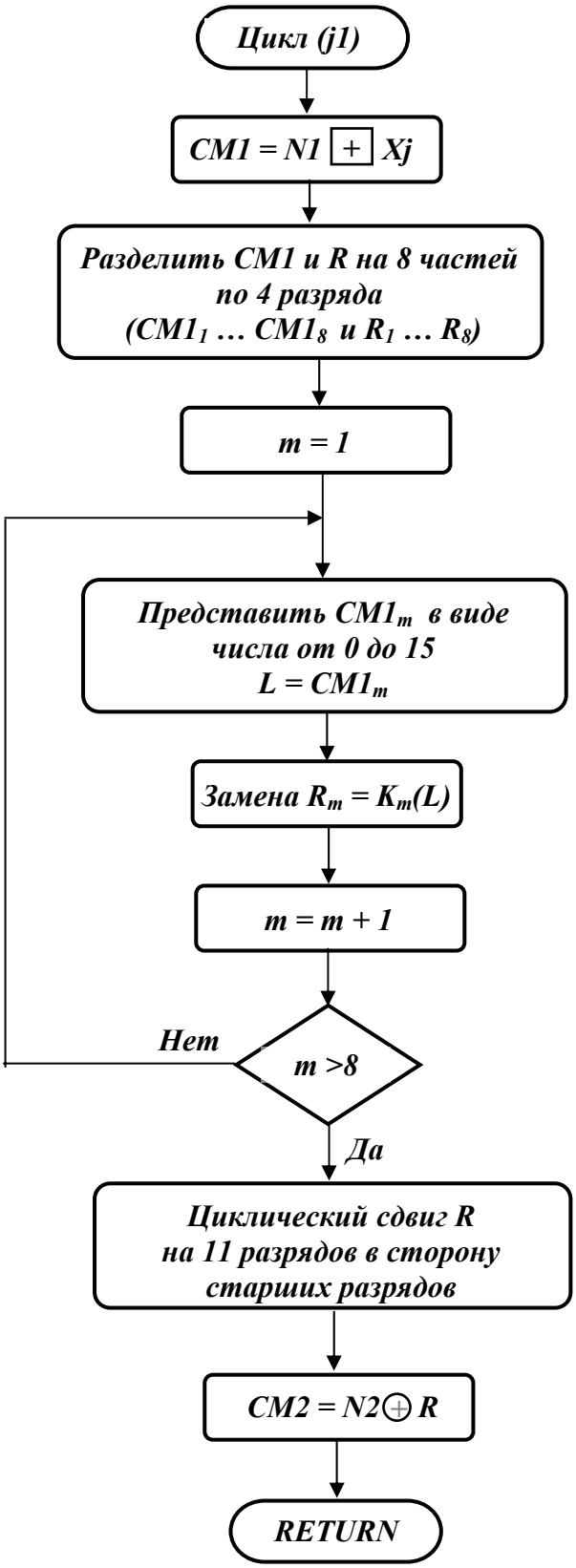
2. Константа С2 имеет вид

Разряд накопителя N5	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
Значение разряда	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Разряд накопителя N5	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Значение разряда	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

ПРИЛОЖЕНИЕ 3
(Справочное)

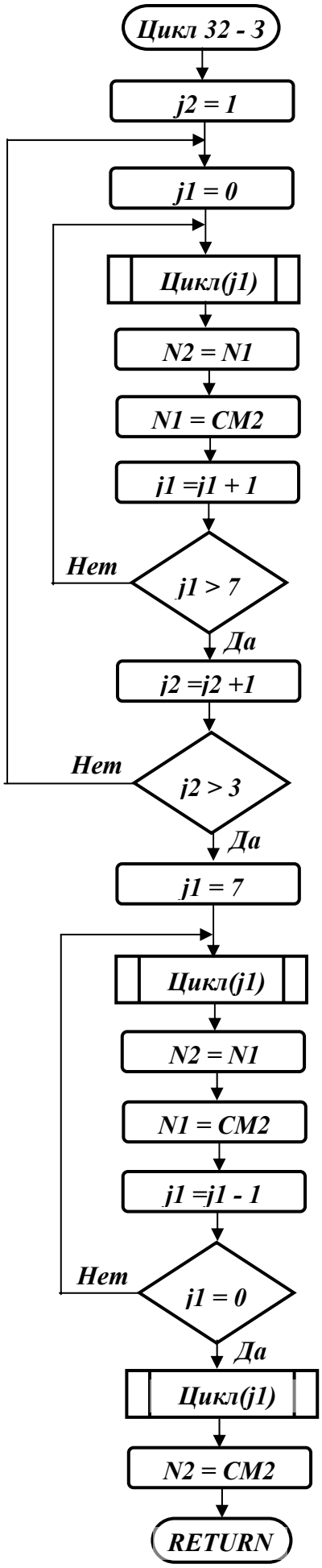
СХЕМЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ АЛГОРИТМА КРИПТОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ

1. Схема одного цикла шифрования



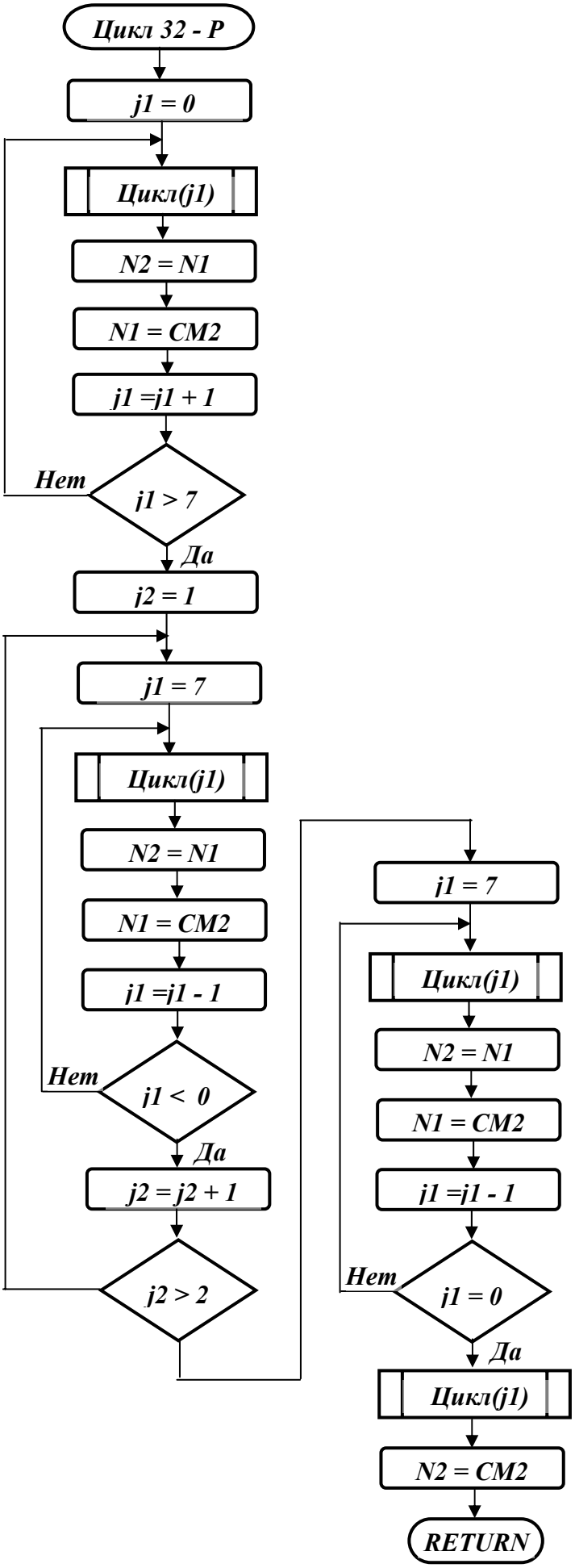
Черт. 5

2. Схема 32-х циклов зашифрования



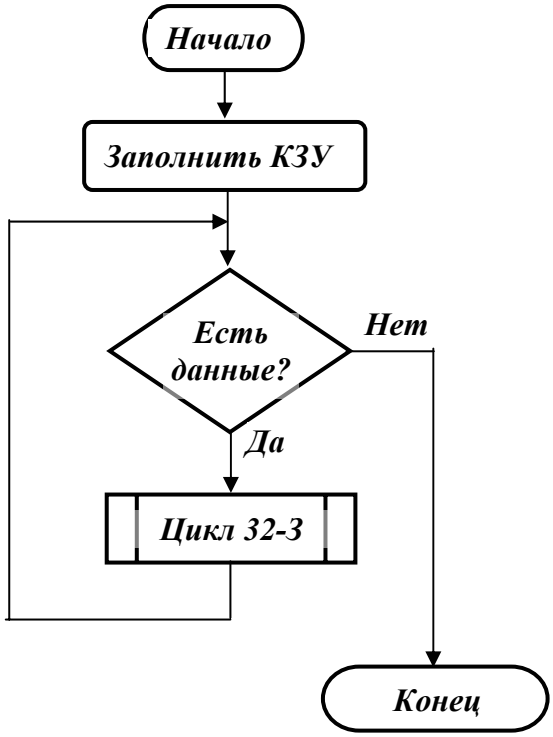
Черт. 6

3. Схема 32-х циклов расшифрования



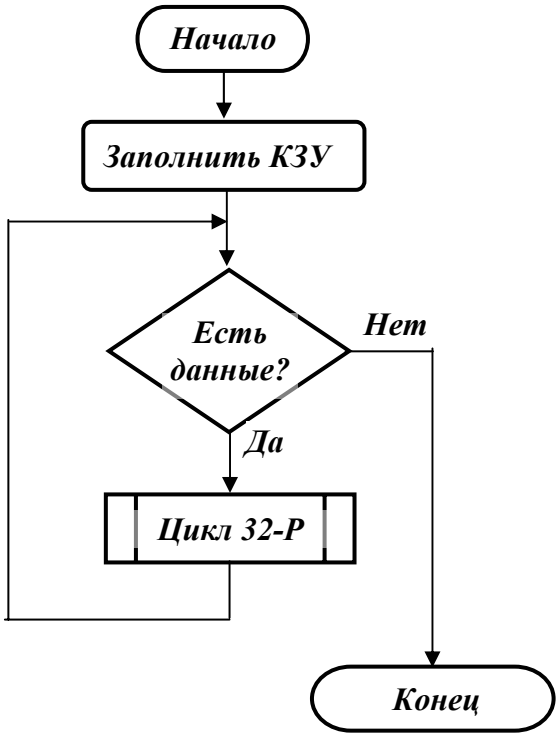
Черт. 7

4. Схема алгоритма зашифрования в режиме простой замены



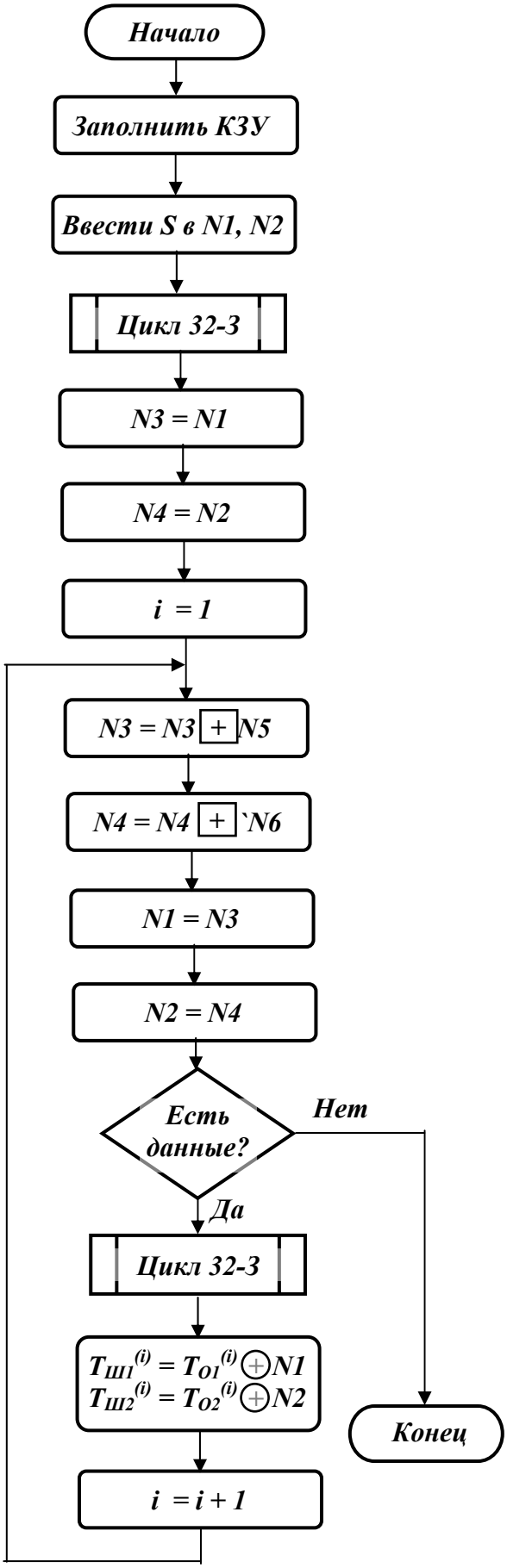
Черт. 8

5. Схема алгоритма расшифрования в режиме простой замены



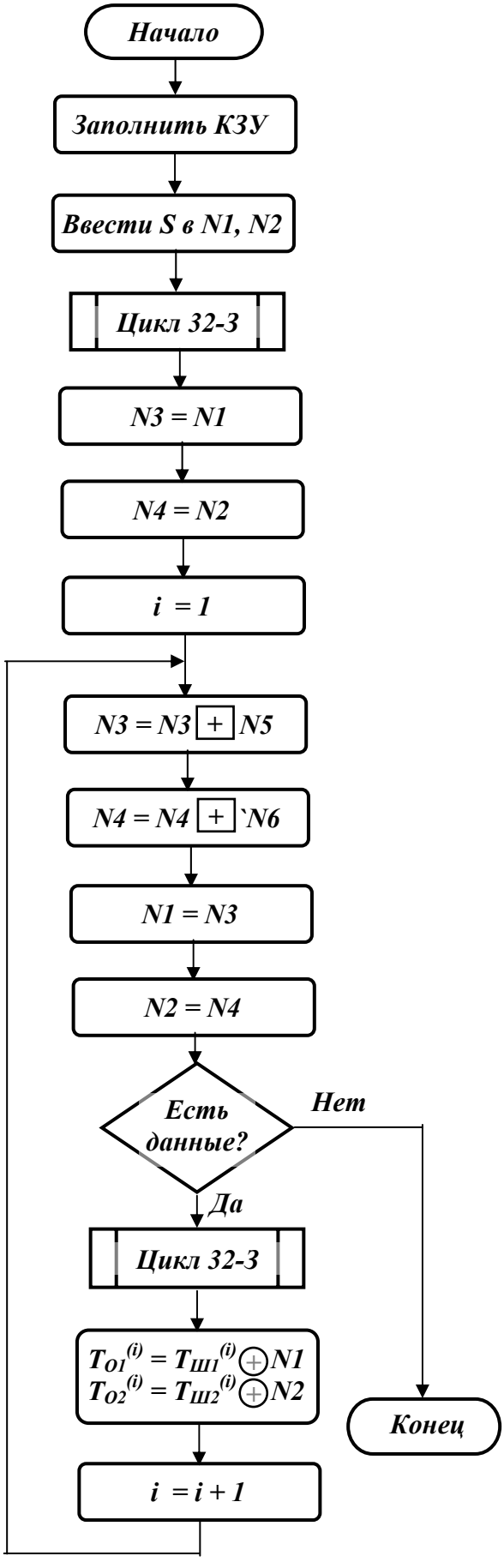
Черт. 9

6. Схема алгоритма зашифрования в режиме гаммирования



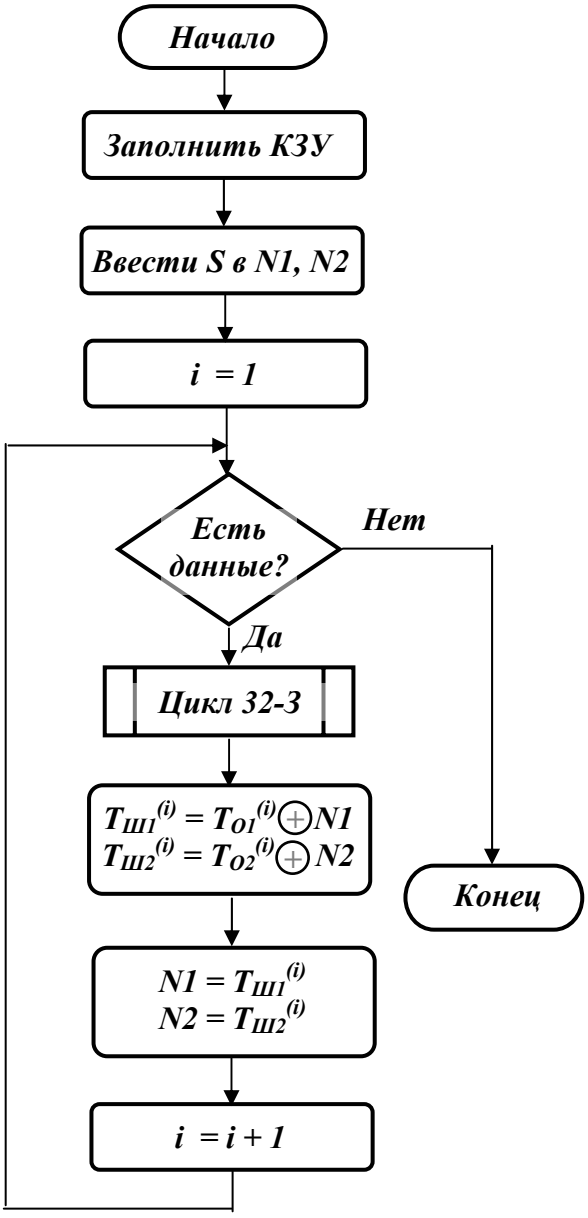
Черт. 10

7. Схема алгоритма расшифрования в режиме гаммирования



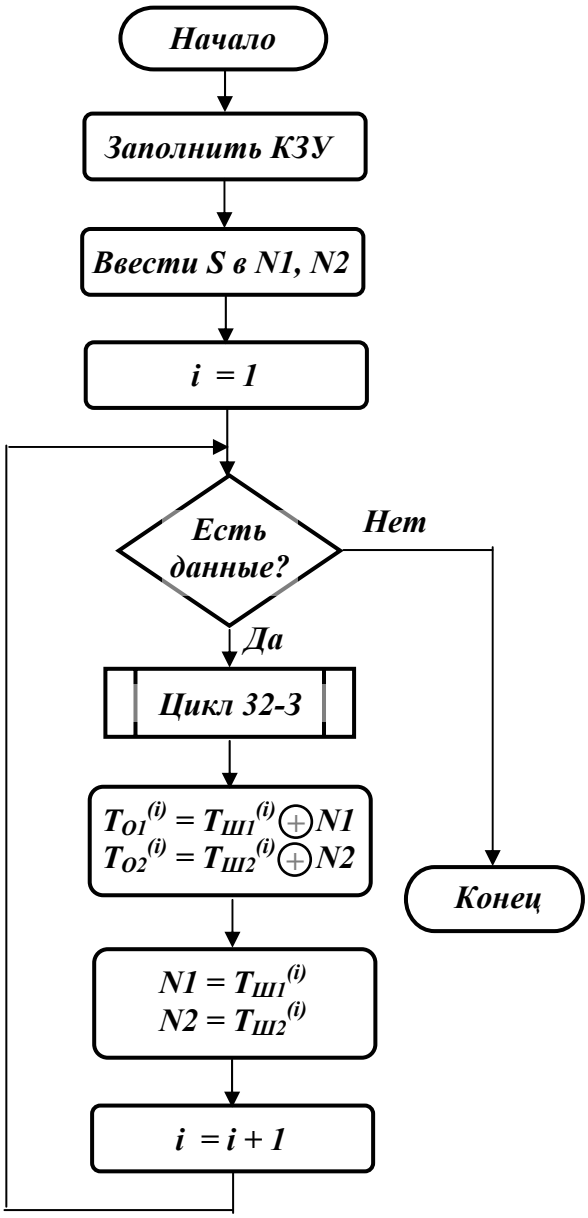
Черт. 11

8. Схема алгоритма зашифрования в режиме гаммирования с обратной связью



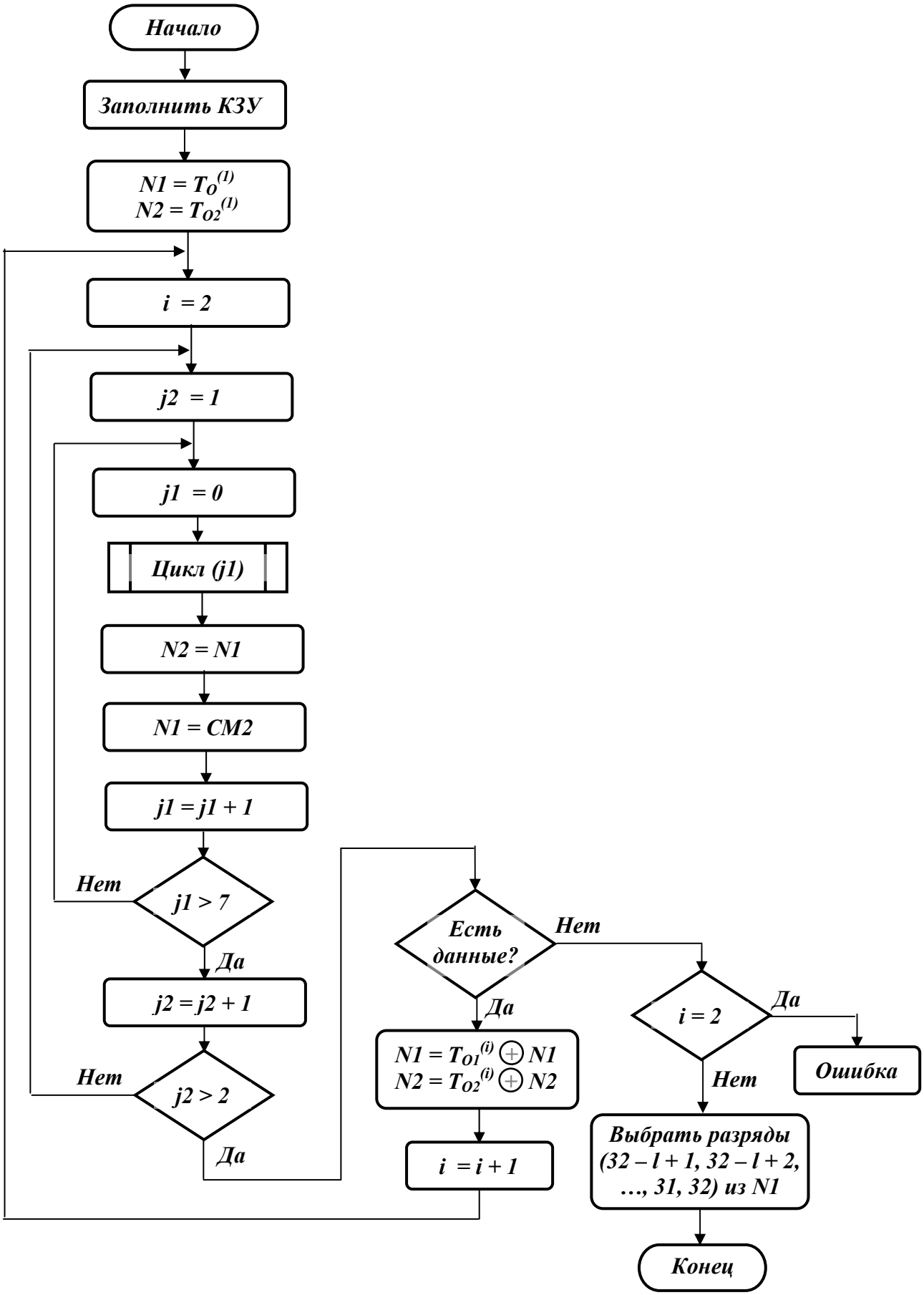
Черт. 12

9. Схема алгоритма расшифрования в режиме гаммирования с обратной связью



Черт. 13

9. Схема алгоритма криптографического преобразования в режиме выработки имитовставки



Черт. 14

ПРИЛОЖЕНИЕ 4
(Справочное)

ПРАВИЛА СУММИРОВАНИЯ ПО МОДУЛЮ 2^{32} И ПО МОДУЛЮ $(2^{32} - 1)$

1. Два целых числа **a**, **b**, где $0 \quad a, b \quad 2^{32} - 1$, представленные в двоичном коде

$$a = (a_{32}, a_{31}, \ldots, a_2, a_1), b = (b_{32}, b_{31}, \ldots, b_2, b_1),$$ т. е.

$$a = a_{32} \quad 2^{31} + a_{31} \quad 2^{30} \ldots + a_2 \quad 2 + a_1, \quad b = b_{32} \quad 2^{31} + b_{31} \quad 2^{30} \ldots + b_2 \quad 2 + b_1,$$

суммируются по модулю 2^{32} (операция $\boxed{+}$) по следующему правилу:

$$a \boxed{+} b = a + b, \text{ если } a + b < 2^{32},$$

$$a \boxed{+} b = a + b - 2^{32}, \text{ если } a + b \quad 2^{32},$$

где операция **+ (-)** есть арифметическая сумма (разность) двух целых чисел.

2. Два целых числа **a**, **b**, где $0 \quad a, b \quad 2^{32} - 1$, представленные в двоичном коде

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

суммируются по модулю $(2^{32} - 1)$ (операция $\boxed{+}$`) по следующему правилу:

$$a \boxed{+}` b = a + b, \text{ если } a + b < 2^{32},$$

$$a \boxed{+}` b = a + b - 2^{32} + 1, \text{ если } a + b \quad 2^{32}.$$

ИНФОРМАЦИОННЫЕ ДАННЫЕ

1. УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Государственного комитета СССР по стандартам от 02.06.89 №1409.

2. ВВЕДЕН ВПЕРВЫЕ.

3. ССЫЛОЧНЫЕ НОРМАТИВНО - ТЕХНИЧЕСКИЕ ДОКУМЕНТЫ

Обозначение НТД, на который дана ссылка	Номер пункта
ГОСТ 15971-90	Приложение 1
ГОСТ 17657-79	Приложение 1
ГОСТ 19781-90	Приложение 1

4. Переиздание, апрель 1996 г.