Часть II. Хеширование: вычисление сжатого образа сообщения

Функция хеширования (кратко: *хеш-функция*) H – это криптографическая функция (процедура или алгоритм), применяемая к сообщению M произвольной длины m и возвращающая битовое значение (*хеш-значение*) $h = \mathrm{H}(M)$ фиксированной длины n, называемое *сверткой*, или *сжатым образом сообщения*.

Хеш-функция H должна обладать следующими свойствами:

1. (***Односторонность***.) Для заданного h задача нахождения сообщения M, для которого $H(M) = h$, должна быть вычислительно-трудоемкой (в то время как значение $H(M)$ вычисляется относительно легко для любого M).
2. (***Слабая сопротивляемость коллизиям***.) При заданном M задача нахождения другого сообщения M', для которого $H(M') = H(M)$, должна быть вычислительно-трудоемкой. (Пара сообщений M и M' образуют *коллизию*, если $H(M') = H(M)$.) В некоторых приложениях необходимо выполнение дополнительного свойства:
3. (***Сильная сопротивляемость коллизиям***.*)* Задача нахождения двух случайных сообщений M и M', для которых $H(M') = H(M)$, должна быть вычислительно-трудоемкой (с экспоненциальным объемом перебора).

Хеш-функции, обладающие указанными свойствами, используются при решении следующих криптографических задач:

- контроль целостности данных при их передаче и хранении;
- аутентификация источника данных.

При решении первой задачи для каждого сообщения (набора данных) M вычисляется его свертка $h = H(M)$, которая передается и хранится вместе с сообщением как контрольное значение. При получении данных получатель вычисляет значение свертки и сравнивает его с имеющимся контрольным значением. Несовпадение означает, что данные были изменены. Чтобы противник (злоумышленник) не смог самостоятельно вычислить контрольное значение свертки и тем самым осуществить необнаруживаемую подмену данных, свертка должна зависеть от секретного параметра (ключа), известного только отправителю и получателю сообщения. Хеш-функции, зависящие от секретного ключа, называют *ключевыми*, или *кодами аутентификации сообщений* ($MAC - Message\ Autentication\ Code$), другие названия – *имитовставка, криптографическая контрольная сумма.*

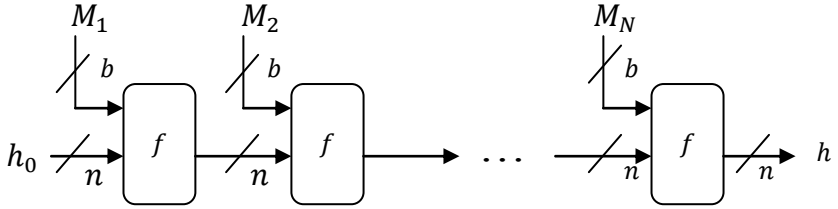
При решении второй задачи – аутентификация источника сообщения – предполагают, что стороны (отправитель и получатель) не доверяют друг другу. Поэтому способ аутентификации, основанный на использовании общей секретной ключевой информации, как в предыдущем случае, здесь неприменим. В данном случае аутентификация осуществляется с помощью цифровой подписи. Обычно подписывается не само сообщение, а его свертка (сжатый образ), вычисленная с помощью хеш-функции. При этом способ вычисления свертки не является секретным (но предполагается, что подбор двух сообщений с одинаковым значением свертки является трудной задачей). Хеш-функции, не зависящие от секретного ключа, называют *бесключевым*, или *кодами обнаружения ошибок* ($MDS - Manipulation\ Detection\ Code$).

Как правило, хеш-функцию строят на основе т.н. *одношаговых сжимающих функций* $y = f(x_1, x_2)$, где $x_1 - b$-битовая, а $x_2 - n$-битовая переменные, возвращаемое значение y является -битовым блокам, причем n – длина свертки. (Обычно $b > n$, поэтому соответствующие функции f называют *сжимающими.*) Для получения значения $h = H(M)$ сообщение M предварительно дополняется битовой комбинацией $10 \ldots 0$ и 64-битовой записью длины m исходного сообщения так, чтобы длина сообщения

, включая дополнение, была кратна b. Затем сообщение разбивается на -битовые блоки M_1,M_2,……, M_Nи к ним применяется следующая итерационная схема вычисления свертки (см. рис. 1):

$h := h_0$;
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ N\ \boldsymbol{do}\ h := f(M_i, h)$.

Здесь h_0 – некоторое фиксированное начальное -битовое значение (его называют *вектором инициализации* хеш-функции).



h_0 – начальное значение
M_i – вводимый блок
f – функция сжатия
n – длина хеш-значения
b – длина вводимого блока

Рис.1. *Общая структура хеш-функции*

II.1. Ключевые функции хеширования

Ключевая хеш-функция может быть построена на основе алгоритма блочного шифрования. Пусть $\mathcal{E}_k$ – функция зашифрования -битового блока под управлением секретного ключа k. Тогда значение n-битовой свертки $h = H(M)$ может быть определено как

$h := 0$;
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ N\ \boldsymbol{do}\ h := \mathcal{E}_k(M_i \oplus h)$.

Данный способ вычисления хеш-значения в российском стандарте криптографической защиты данных ГОСТ 28147-89 называется *режимом выработки имитовставки*.

Ключевая хеш-функция может быть построена также на основе бесключевой хеш-функции $H(M)$. Секретный ключ k, дополненный некоторым способом до размера, кратного длине блока n, вставляется в начало и конец сообщения M. Значение h определяется как

$$h := H(k||M||k).$$

Заметим, что ключ k непосредственно не пересылается, а присоединяется к M только на время вычисления значения h.)

Другой способ вычисления хеш-значения h определяется как

$$h := H(k||h_1), \text{ где } h_1 = H(k||M).$$

II.2. Бесключевые функции хеширования

Бесключевую хеш-функцию можно построить, используя симметричный блочный шифр. Пусть $\mathcal{E}_k$ – функция зашифрования -битовых блоков под управлением n-битового ключа k, а M – сообщение, представленное в виде последовательности n

-битовых блоков M_1,M_2,……, M_N. Схема вычисления -битового хеш-значения $h = H(M)$ имеет следующий вид (см. рис.2):

$h := h_0$;
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ N\ \boldsymbol{do}\ h := \mathcal{E}_A(B) \oplus C$,

где A, B, C могут принимать значения M_i, h, $M_i \oplus h$ или быть константами. Существует $4^3 = 64$ комбинации для выбора переменных A, B, C, но установлено, что только 12 комбинаций, перечисленных в таблице 1, приводят к безопасным хеш-функциям. Первые четыре из этих схем иллюстрируются на рис.3.

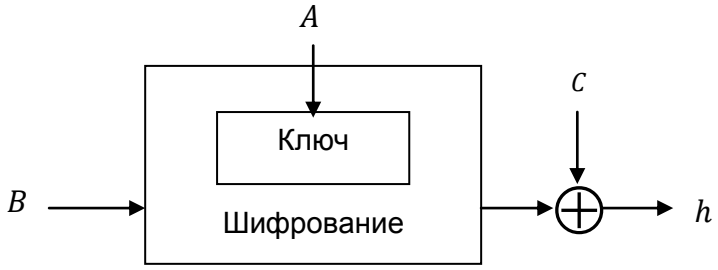


Рис.2. *Обобщенная схема хеш-функции на основе блочного шифра с размером хеш-значения, равным размеру блока*

Таблица 1. *Параметры схем безопасного хеширования (к рис.2)*

Номер схемы	A	B	C	Номер схемы	A	B	C
1	h	M_i	M_i	7	M_i	h	$M_i \oplus h$
2	h	$M_i \oplus h$	$M_i \oplus h$	8	M_i	$M_i \oplus h$	h
3	h	M_i	$M_i \oplus h$	9	$M_i \oplus h$	M_i	M_i
4	h	$M_i \oplus h$	M_i	10	$M_i \oplus h$	h	h
5	M_i	h	h	11	$M_i \oplus h$	M_i	h
6	M_i	$M_i \oplus h$	$M_i \oplus h$	12	$M_i \oplus h$	h	M_i

Для противодействия атакам на хеш-функцию необходимо, чтобы длина вырабатываемого хеш-значения составляла по меньшей мере 128 битов. Увеличить размер хеш-значения можно, используя, например, следующий прием:

1. Вычисляется -битовое хеш-значение $h_1 = H(M)$.
2. Значение h_1 приписывается в начало сообщения M и вычисляется новое хеш-значение $h_2 = H(h_1||M)$.
3. Этап 2 повторяется для конкатенации $h_2||M$ и т.д.

В результате получается последовательность h_1, h_2,…, h_s n-битовых хеш-значений, конкатенация которых $h_1||h_2||\ ...\ ||h_s$ дает хеш-значение длины sn. Формально данная схема определяется как:

$$h := H(M);\ g := h;\ \boldsymbol{for}\ i := 2\ \boldsymbol{to}\ S\ \boldsymbol{do}\ \{g := H(g||M);\ h := g||h\}.$$

Из -битового значения h можно извлечь хеш-значение требуемой длины $t \le sn$, отбрасывая, например, последние $sn - t$ битов. Другие способы увеличения длины хеш-значения (в 2 раза) представлены на рис.4, 5.

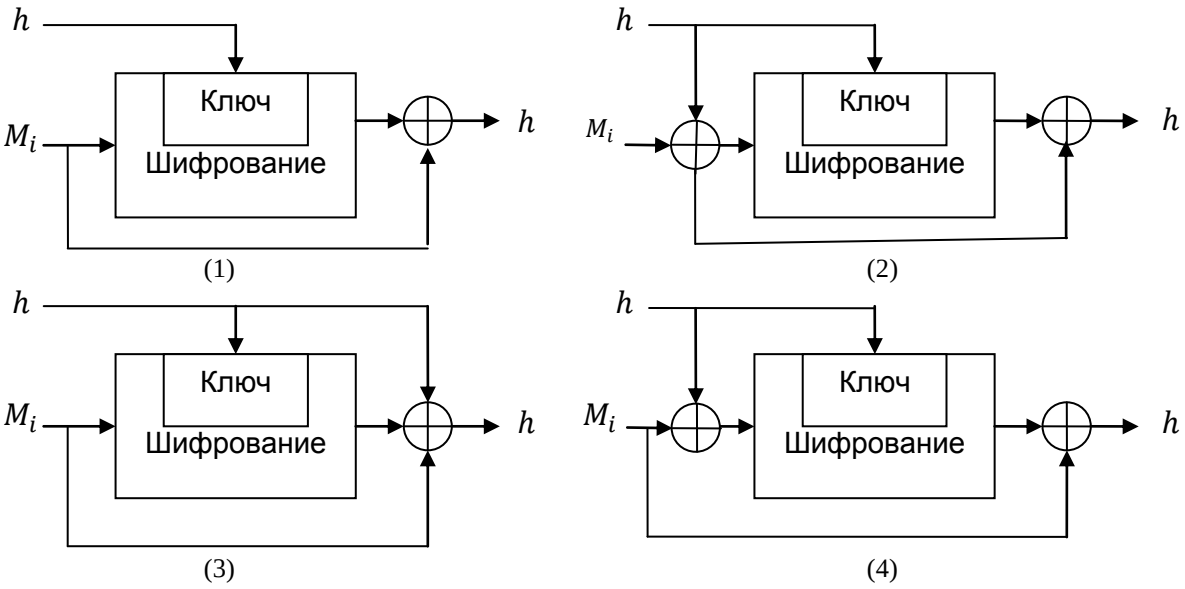
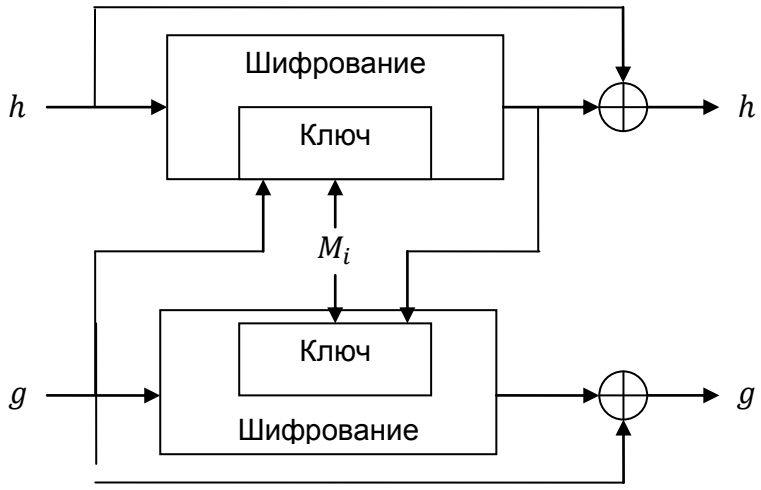
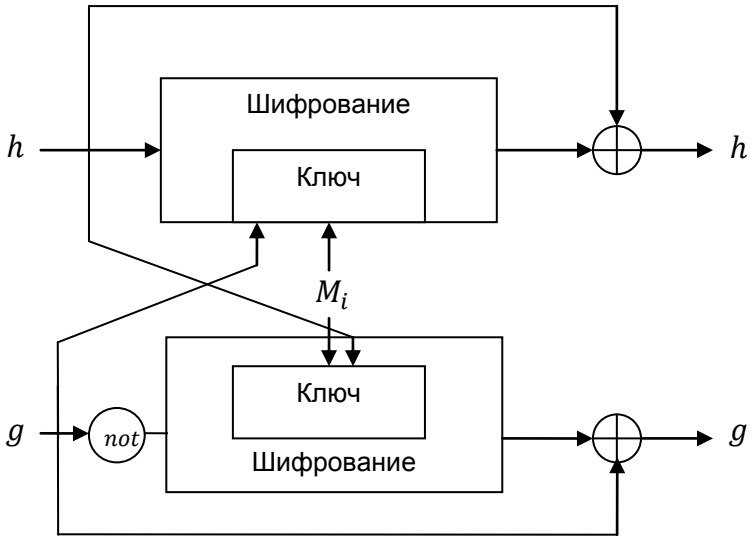


Рис.3. *Четыре безопасные хеш-функции с размером хеш-значения, равным блоку*



(*a*)

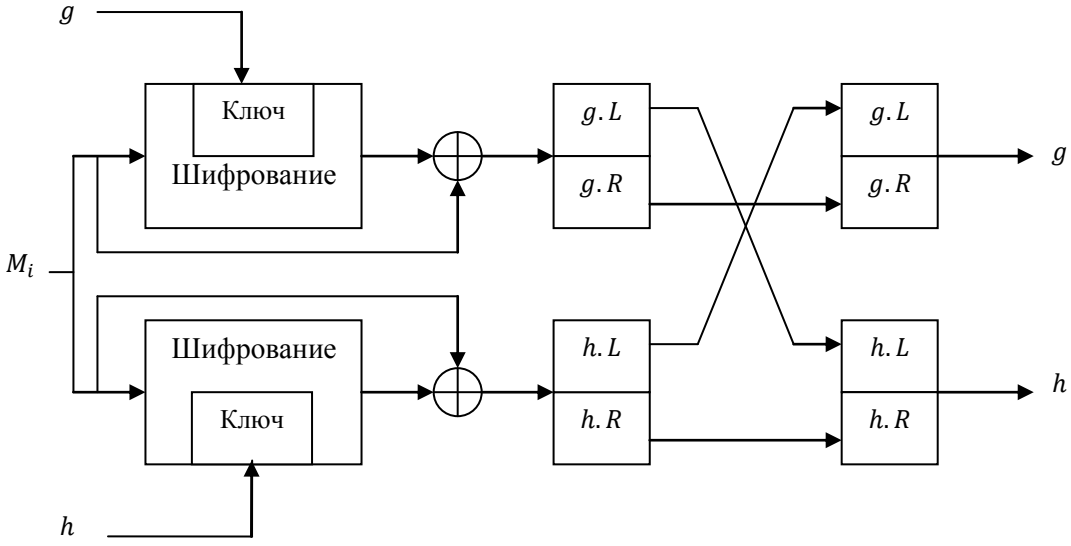
$h := h_0;\ g := g_0;$
for $i := 1$ ***to*** N ***do*** {
$w := \mathcal{E}_{g||M_i}(h);$
$g := g \oplus \mathcal{E}_{M_i||w}(g);\ h := w \oplus h$
}.



(*б*)

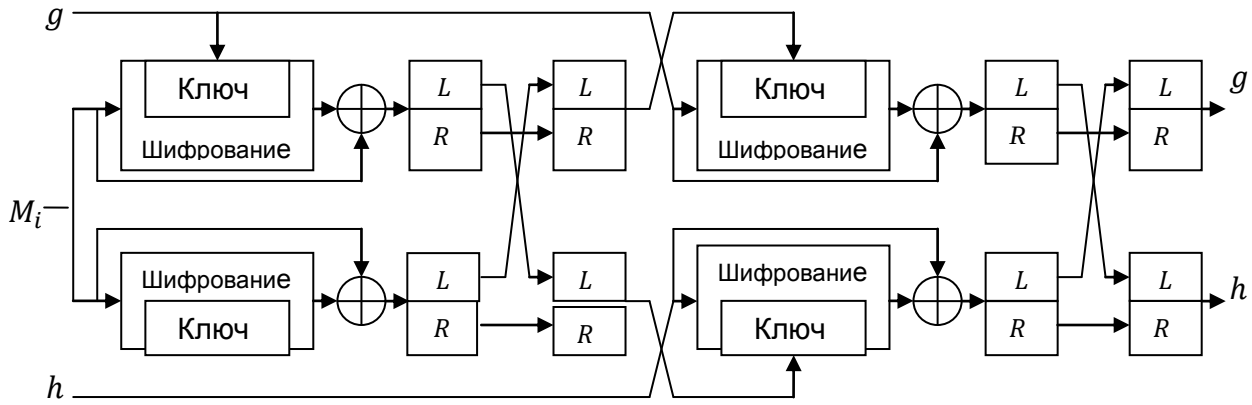
$h := h_0;\ g := g_0;$
for $i := 1$ ***to*** N ***do*** {
$w := g \oplus \mathcal{E}_{M_i||h}(not\ g); h := h \oplus \mathcal{E}_{g||M_i}(h);$
$g := w$
}.

Рис.4. *Схемы вычисления* 128*-битового хеш-значения* $h||g$ *с использованием* 64*-битовой функции шифрования* $\mathcal{E}_k$ *под управлением* 64*-битового ключа* k



(a)

$h := h_0;\ g := g_0;$
for $i := 1$ ***to*** N ***do*** {
$g := \mathcal{E}_g(M_i) \oplus M_i;\ h := \mathcal{E}_h(M_i) \oplus M_i;$
$g.L \leftrightarrow h.L$
}.



(*б*)

$h := h_0;\ g := g_0;$
for $i := 1$ ***to*** N ***do*** {
$w_g := g;\ w_h := h;$
$g := \mathcal{E}_g(M_i) \oplus M_i;\ h := \mathcal{E}_h(M_i) \oplus M_i;$
$g.L \leftrightarrow h.L;$
$g := \mathcal{E}_g(w_g) \oplus w_g;\ \ h := \mathcal{E}_h(w_h) \oplus w_h;$
$g.L \leftrightarrow h.L$
}.

Рис.5. *Схемы* (*а*) МДС-2 *и* (*б*) МДС-4 *вычисления* $2n$*-битового хеш-значения* $g||h$ *с использованием -битовой функции шифрования* $\mathcal{E}_k$ *под управлением* n*-битового ключа*

II.3. Некоторые алгоритмы хеширования

MD2

Хеш-функция $MD2$ (аббревиатура MD образована от $Message\ Digest$ – дайджест сообщения), автором которой является $Ronald\ Rivest$ (США), возвращает для любого входного сообщения 128-битовое (16-байтовое) хеш-значение.

В алгоритме используется подстановка $S = (S_0, S_1,\ldots, S_t)$, заданная на множестве байтов. Эта подстановка (на ней базируется стойкость функции $MD2$) фиксирована и генерируется на основе числа π.

Алгоритм состоит из следующих шагов:

1. Сообщение M дополняется байтами так, чтобы размер сообщения стал кратен 16 байтам.
2. К сообщению M добавляется 16 байтов контрольной суммы. Сообщение M разбивается на 16-байтовые блоки.
3. Инициализируется 48-байтовый блок: $X_0, X_1,\ldots, X_{47}$. Первые 16 байтов $X_0, X_1,\ldots, X_{15}$ обнуляются.
4. Очередной 16-байтовый блок сообщения M записывается в блок $X_{16}, X_{17},\ldots, X_{31}$. В блок $X_{32}, X_{33},\ldots, X_{47}$ записывается побитовая сумма по модулю 2 блоков $X_0, X_1,\ldots, X_{15}$ и $X_{16}, X_{17},\ldots, X_{31}$:

   for $i := 0$ ***to*** 15 ***do*** $X_{32+i} := X_i \oplus X_{16+i}$.
5. К блоку $X_0, X_1,\ldots, X_{47}$ применяется сжимающая функция:

   $t := 0;$
   for $j := 0$ ***to*** 17 ***do*** {
   for $k := 0$ ***to*** 47 ***do*** {
   $t := X_k \oplus S_t;$
   $X_k := t;$

$t := (t + j) \; mod \; 256$

}

}.

6. Шаги 4 и 5 повторяются, пока не будут исчерпаны все блоки сообщения $\boldsymbol{M}$. Результатом хеширования является 16-байтовый блок X_0, X_1,…, X_{15}.

$\boldsymbol{MD5}$

Хеш-функция $MD5$, автором которой является, как и для $MD2$, $Ronald\ Rivest$, возвращает для сообщения M произвольной длины 128-битовое хеш-значение $h = H(M)$.

В алгоритме используются следующие операции, выполняемые над 32-битовыми блоками: сложение по модулю 2^{32} $(+)$, побитовые операции – отрицание (¬, или not), конъюнкция (&, или and), дизъюнкция (∨, или or), сложение по модулю 2 (⊕, или xor), циклический сдвиг влево на s битовых позиций (rol_s). В 32-битовых блоках (словах) используется прямой порядок байтов ($little\ endian$): младший байт занимает младшую адресную позицию.

Описание $MD5$. Оригинальное битовое сообщение M дополняется справа до длины, кратной 512: сначала добавляется битовая единица 1, затем n битовых нулей, наконец, 64-битовое представление числа m. (Нетрудно рассчитать, что n – число добавляемых нулей – равно $447 - r$ или $959 - r$ соответственно для $r \leq 447$ и $r \geq 448$, где $r = m \; mod \; 512$.) Дополненное сообщение состоит из $N = \frac{m+n+65}{512}$ 512-битовых блоков M_1, M_2,…, M_N. Каждый блок M_i представляется в виде массива из шестнадцати 32-битовых подблоков: $M_i = (M_{i,0}, M_{i,1},\ldots, M_{i,15})$.

Вычисляемое 128-битовое хеш-значение $h = H(M)$ представляется в виде массива из четырех 32-битовых блоков: $h = (h_0, h_1, h_2, h_3)$.

Псевдокод алгоритма представлен в таблице 2 (элементарная операция алгоритма иллюстрируется на рис. 6). В описании используется 128-битовая переменная a, представленная в виде массива четырех 32-битовых блоков: $a = (a_0, a_1, a_2, a_3)$. Функции $F_j(x, y, z)$ с 32-битовыми аргументами и значениями определяются как

$$F_1(x, y, z) = (x\&y) \vee ((not\ x)\&\ z),$$
$$F_2(x, y, z) = (x\&z) \vee (y\&(not\ z)),$$
$$F_3(x, y, z) = x \oplus y \oplus z,$$
$$F_4(x, y, z) = y \oplus x \vee (not\ z)).$$

Значения $s(j, k)$ – величины циклических сдвигов – приведены в таблице 3, а значения $r(j, k)$ – индексы, определяющие выбор входных блоков M_{ir} – задаются формулой:

$$r(j,k) = \begin{cases} k \quad \text{для } j = 0, \\ 5k + 1(mod\ 16) \quad \text{для } j = 1, \\ 3k + 5(mod\ 16) \quad \text{для } j = 2, \\ 7k(mod\ 16) \quad \text{для } j = 3. \end{cases}$$

Значения констант C_{jk} приведены в табл. 4 (16-ичная константа C_{jk} образована целой частью числа $2^{32} sin(k + 16j + 1)$).

Таблица 2. *Псевдокод MD5*

$h := (0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476);$
for $i := 1$ ***to*** N ***do*** $\{a := h;$
for $j := 0,1,2,3$ ***do*** $\{$
for $k := 0$ ***to*** 15 ***do***$\{$
$h0 := h1 + rol_{s(j,k)}(h_0 + F_j(h_1, h_2, h_3)M_{i,r(j,k)} + C_{jk});$
$(h_0, h_1, h_2, h_3) := (h_3, h_0, h_1, h_2)$
$\}$
$\};$
$h := (h_0 + a_0, h_1 + a_1, h_2 + a_2, h_3 + a_3)$
$\}.$

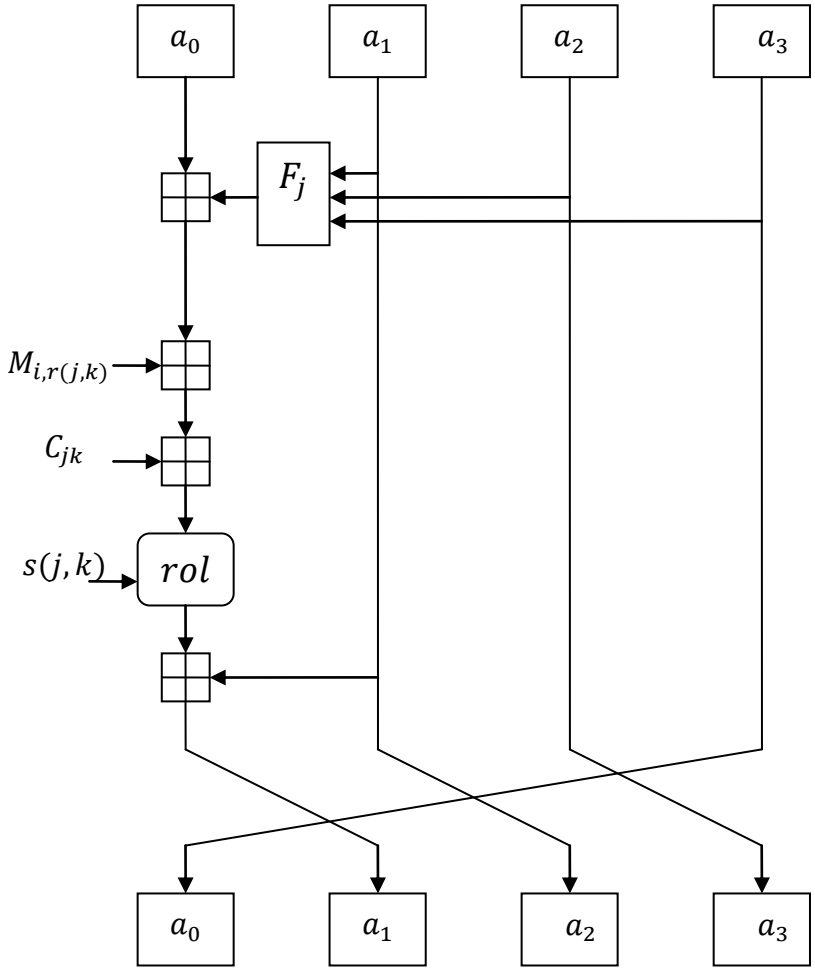


Рис.6. *Элементарная операция MD5*

Таблица 3. *Величины циклических сдвигов в главном цикле MD5*

K	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(0,k)$	7	12	17	22	7	12	17	22	7	12	17	22	7	12	17	22
$s(1,k)$	5	9	14	20	5	9	14	20	5	9	14	20	5	9	14	20
$s(2,k)$	4	11	16	23	4	11	16	23	4	11	16	23	4	11	16	23
$s(3,k)$	6	10	15	21	6	10	15	21	6	10	15	21	6	10	15	21

Таблица 4. *Константы* C_{jk} *в* $MD5$ *(в* 16*-ичном представлении)*

k	C_{0k}	C_{1k}	C_{2k}	C_{3k}
0	$d76aa478$	$f61e2562$	$fffa3942$	$f4292244$
1	$e8c7b756$	$c040b340$	$8771f681$	$432aff97$
2	$242070db$	$265e5a51$	$6d9d6122$	$ab9427a7$
3	$c1bdceee$	$e9b6c7aa$	$fde5380c$	$fc93a039$
4	$f57c0faf$	$d62f105d$	$a4beea44$	$655b59c3$
5	$4787c62a$	02441453	$4bdecfa9$	$8f0ccc92$
6	$a8304613$	$d8a1e681$	$f6bb4b60$	$ffeff47d$
7	$fd469501$	$e7d3fbc8$	$bebfbc70$	85845dd1
8	$698098d8$	$21e1cde6$	$289b7ec6$	$6fa87e4f$
9	$8b44f7af$	$c33707d6$	$eaa127fa$	$fe2ce6e0$
10	$ffff5bb1$	f4d50d87	d4ef3087	a3014314
11	$895cd7be$	$455a14ed$	$04881d05$	$4e0811a1$
12	$6b901122$	$a9e3e905$	$d9d4d039$	$f7537e82$
13	$fd987193$	$fcefa3f8$	$e6db99e5$	$bd3af235$
14	$a679438e$	$676f02d9$	$1fa27cf8$	$2ad7d2bb$
15	$49b40821$	$8d2a4c8a$	$c4ac5665$	$eb86d391$

***RIPEMD*-160**

Хеш-функция $RIPEMD$-160, разработанная по инициативе Европейского Сообщества в рамках проекта $RIPE$ ($Race\ Integrity\ Primitives\ Evaluation$), вычисляет 160-битовые хеш-значения для сообщения произвольной длины.

Описание $RIPEMD$-160. Входное битовое сообщение M дополняется таким же способом, как и в алгоритме $MD5$, и представляется в виде последовательности 512-битовых блоков M_1,…, M_N. Каждый блок M_i представляется в виде массива из шестнадцати 32-битовых подблоков: $M_i = (M_{i,0}, M_{i,1},\ldots, M_{i,15})$. Вычисляемое 160-битовое хеш-значение $h = H(M)$ представляется в виде массива из пяти 32-битовых подблоков: $h = (h_0, h_1, h_2, h_3, h_4)$.

В алгоритме используются такие же операции над 32-битовыми блоками, что и в $MD5$. В 32-битовых блоках (словах) используется прямой порядок байтов ($little\ endian$): младший байт занимает младшую адресную позицию.

Псевдокод алгоритма представлен в табл. 5 (см. рис. 7 и 8). В алгоритме используются вспомогательные 160-битовые переменные a и b, представленные в виде массивов 32-битовых подблоков: $a = (a_0, a_1, a_2, a_3, a_4)$, $b = (b_0, b_1, b_2, b_3, b_4)$. Функции $F_j(x, y, z)$ с 32-битовыми аргументами и значениями определяются следующим образом:

$$F_j(x,y,z) = \begin{cases} x \oplus y \oplus z & \text{для } 0 \le j \le 15, \\ (x \& y) \vee ((\neg x) \& z) & \text{для } 16 \le j \le 31, \\ (x \vee (\neg y)) \oplus z & \text{для } 32 \le j \le 47, \\ (x \& z) \vee (y \& (\neg z)) & \text{для } 48 \le j \le 63, \\ x \oplus (y \vee (\neg z)) & \text{для } 64 \le j \le 79. \end{cases}$$

Используемые в алгоритме константы $C_j^{(a)}$ и $C_j^{(b)}$, значения величин $ra(j)$ и $rb(j)$, $sa(j)$ и $sb(j)$ приведены соответственно в табл. 6-8.

Таблица 5. *Псевдокод RIPEMD*-160

$h := (0x67452301, 0xafcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1fo)$;
for $i := 1$ ***to*** N ***do*** {
$a := h;\ b := h;$
for $j := 0$ ***to*** 79 ***do*** {
(левая линия)
$t := rol_{sa(j)}(a_0 + F_j(a_1, a_2, a_3) + M_{i,ra(j)} + C_j^{(a)}) + a_4;$
$(a_0, a_1, a_2, a_3, a_4) := (a_4, t, a_1, a_2, a_3);$
(правая линия)
$t := rol_{sb(j)}(b_0 + F_{79-j}(b_1, b_2, b_3) + M_{i,rb(j)} + C_j^{(b)}) + b_4;$
$(b_0, b_1, b_2, b_3, b_4) := (b_4, t, b_1, b_2, b_3)$
};
$h := (h_1 + a_2 + b_3, h_2 + a_3 + b_4, h_3 + a_4 + b_0, h_4 + a_0 + b_1, h_0 + a_1 + b_2)$
}.

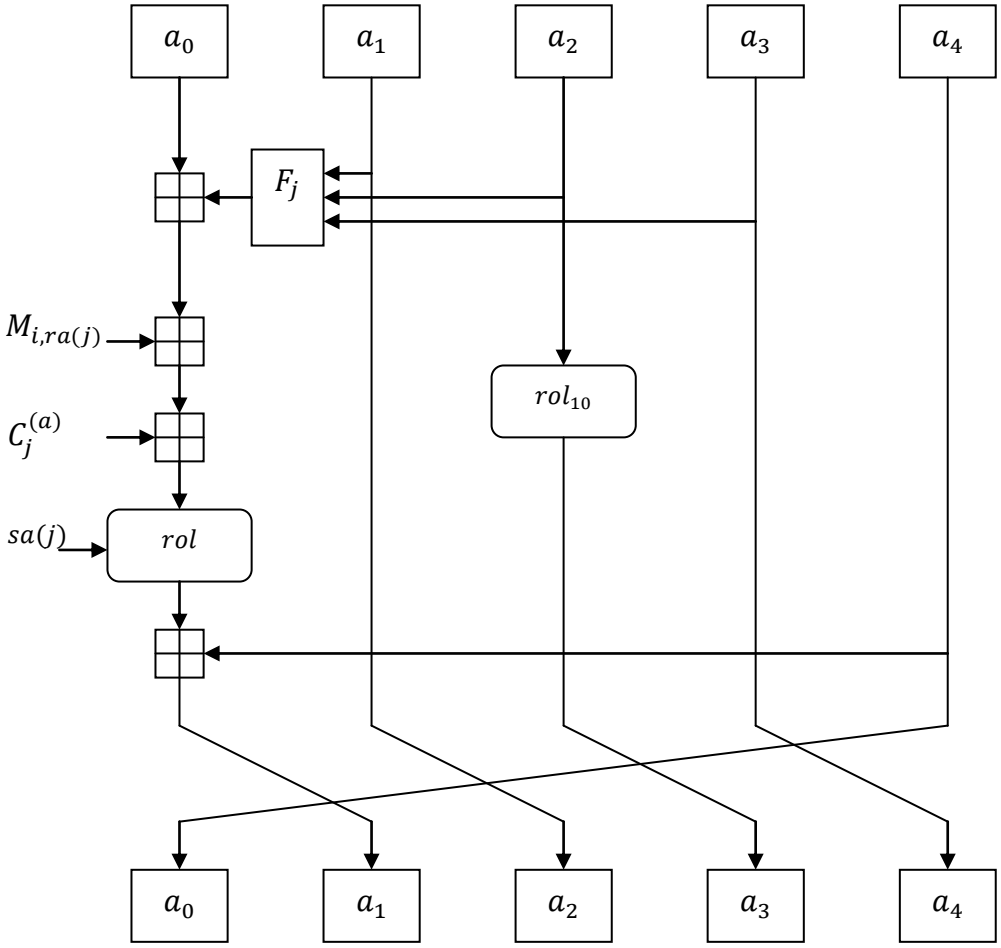


Рис.7. *Элементарная операция RIPEMD*-160 (*левая линия*)

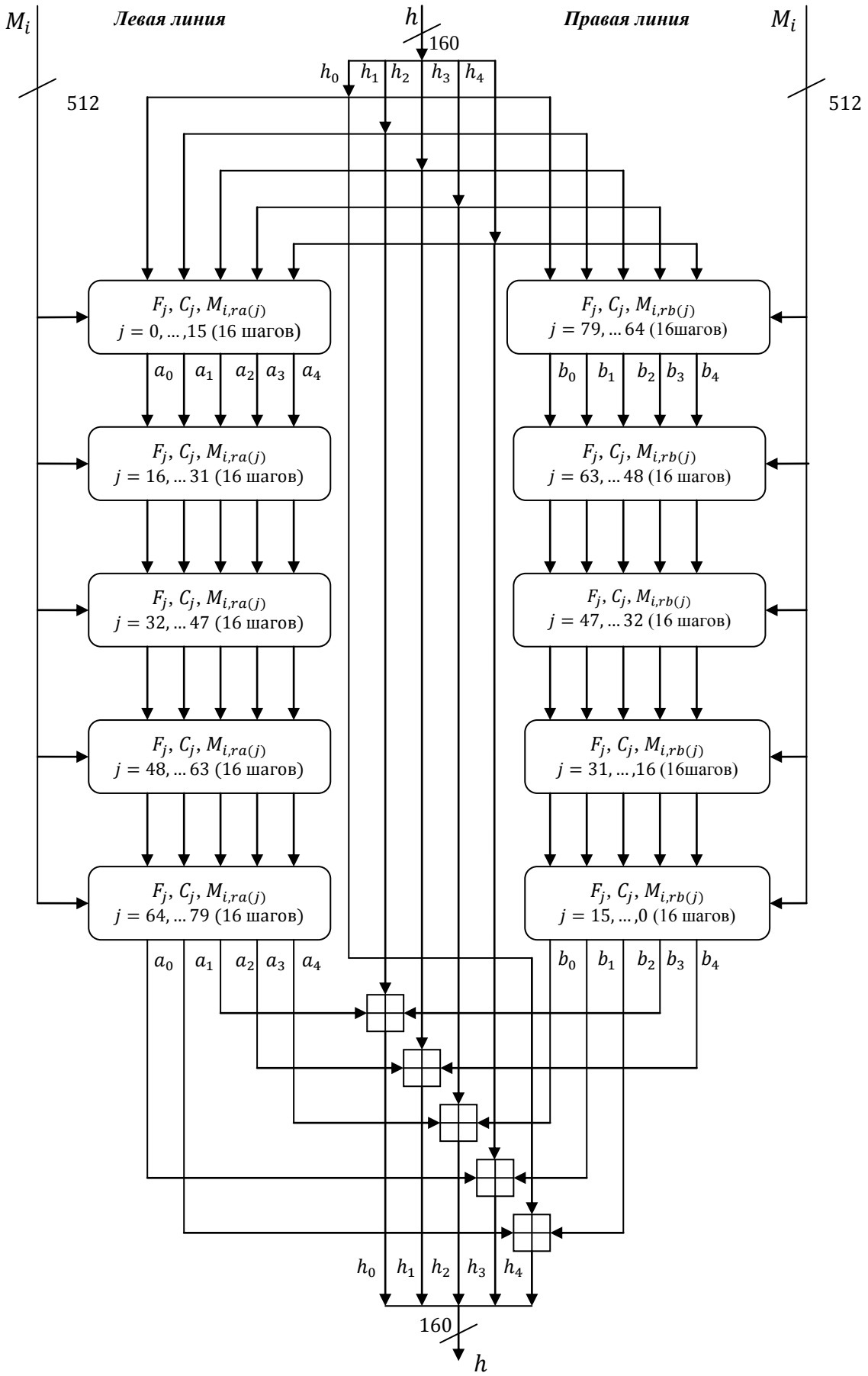


Рис. 8. *Функция сжатия в RIPEMD*-160 (*обработка одного* 512-*битового блока*)

Таблица 6. *Константы $RIPEMD$-160 $C_j^{(a)}$ и $C_j^{(b)}$ в 16-ичном представлении*

Номер шага	Левая линия		Правая линия	
	$C_j^{(a)}$	Целая часть числа	$C_j^{(b)}$	Целая часть числа
$0 \le j \le 15$	00000000	0	$be50a286$	$2^{30}\sqrt[3]{2}$
$16 \le j \le 31$	$5a827999$	$2^{30}\sqrt{2}$	$5c4dd124$	$2^{30}\sqrt[3]{3}$
$32 \le j \le 47$	$6ed9eba1$	$2^{30}\sqrt{3}$	$6d703ef3$	$2^{30}\sqrt[3]{5}$
$48 \le j \le 63$	$8f1bbcdc$	$2^{30}\sqrt{5}$	$7a6d76e9$	$2^{30}\sqrt[3]{7}$
$64 \le j \le 79$	$a953fd4e$	$2^{30}\sqrt{7}$	00000000	0

Таблица 7. *Порядок выбора входных блоков в RIPEMD-160*

левая линия

$ra(0 \dots 15)$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$ra(16 \dots 31)$	7	4	13	1	10	6	15	3	12	0	9	5	2	14	11	8
$ra(32 \dots 47)$	3	10	14	4	9	15	8	1	2	7	0	6	13	11	5	12
$ra(48 \dots 63)$	1	9	11	10	0	8	12	4	13	3	7	15	14	5	6	2
$ra(64 \dots 79)$	4	0	5	9	7	12	2	10	14	1	3	8	11	6	15	13

правая линия

$rb(0 \dots 15)$	5	14	7	0	9	2	11	4	13	6	15	8	1	10	3	12
$rb(16 \dots 31)$	6	11	3	7	0	13	5	10	14	15	8	12	4	9	1	2
$rb(32 \dots 47)$	15	5	1	3	7	14	6	9	11	8	12	2	10	0	4	13
$rb(48 \dots 63)$	8	6	4	1	3	11	15	0	5	12	2	13	9	7	10	14
$rb(64 \dots 79)$	12	15	10	4	1	5	8	7	6	2	13	14	0	3	9	11

Таблица 8. *Величины циклических сдвигов в $RIPEMD$-160*

левая линия

$sa(0 \dots 15)$	11	14	15	12	5	8	7	9	11	13	14	15	6	7	9	8
$sa(16 \dots 31)$	7	6	8	13	11	9	7	15	7	12	15	9	11	7	13	12
$sa(32 \dots 47)$	11	13	6	7	14	9	13	15	14	8	13	6	5	12	7	5
$sa(48 \dots 63)$	11	12	14	15	14	15	9	8	9	14	5	6	8	6	5	12
$sa(64 \dots 79)$	9	15	5	11	6	8	13	12	5	12	13	14	11	8	5	6

правая линия

$sb(0 \dots 15)$	8	9	9	11	13	15	15	5	7	7	8	11	14	14	12	6
$sb(16 \dots 31)$	9	13	15	7	12	8	9	11	7	7	12	7	6	15	13	11
$sb(32 \dots 47)$	9	7	15	11	8	6	6	14	12	13	5	14	13	13	7	5
$sb(48 \dots 63)$	15	5	8	11	14	14	6	14	6	9	12	9	12	5	15	8
$sb(64 \dots 79)$	8	5	12	9	12	5	14	6	8	13	6	5	15	13	11	11

SHA-1

Алгоритм SHA-1 ($Secure\ Hast\ Algorith$) стандарта США на хеш-функцию SHS ($Secure\ Hash\ Standard$) вычисляет 160-битовые хеш-значения для любого сообщения, длина которого меньше 2^{64} битов. Вычисленное хеш-значение используется затем в алгоритме DSA ($Digital\ Signature\ Algorith$), стандарта DSS ($Digital\ Signature\ Standard$), вычисляющем цифровую подпись сообщения.

Оригинальное сообщение M дополняется до длины, кратной 512. Способ дополнения такой же, как и в $MD5$: сначала добавляется битовая комбинация 10 ... 0 так, чтобы общая длина в битах была сравнима с 448 по модулю 512 (операция дополнения выполняется всегда, даже если сообщение уже имело требуемую длину); затем к сообщению добавляется 64-битовое представление длины исходного сообщения.

Замечание. В отличие от $MD5$ и $RIPEMD$-160 в SHA-1 используется обратный порядок байтов ($big\ endian$): наиболее значимый байт идет первым, т.е. занимает младшую

адресную позицию. Это относится как к 64-битовому представлению длины исходного сообщения, так и к 32-битовым числам.

Дополненное сообщение разбивается на 512-битовые блоки M_1,…, M_N. Каждый блок M_i представляется в виде массива из шестнадцати 32-битовых подблоков: $M_i = (M_{i,0}, M_{i,1},\ldots, M_{i,15})$. Вычисляемое хеш-значение $h = H(M)$ представляется в виде массива из пяти 32-битовых подблоков: $h = (h_0, h_1, h_2, h_3, h_4)$.

Псевдокод алгоритма SHA-1 представлен в табл. 9 (см. рис. 9 и 10). В алгоритме используются вспомогательные переменные: $a = (a_0, a_1, a_2, a_3, a_4)$ – массив из пяти 32-битовых подблоков, $W = (w_0, w_1, \ldots, w_{79})$ – массив из 80 32-битовых подблоков. Основные операции над 32-битовыми подблоками те же, что и в алгоритме $MD5$. Функции $F_j(x, y, z)$ с 32-битовыми аргументами и значениями и 32-битовые константы C_j определяются как

$$F_j(x,y,z) = \begin{cases} (x \,\&\, y) \vee ((\neg x) \,\&\, z) & \text{для } 0 \le j \le 19, \\ x \oplus y \oplus z & \text{для } 20 \le j \le 39 \text{ и } 60 \le j \le 79, \\ (x \,\&\, y) \vee (x \,\&\, z) \vee (y \,\&\, z) & \text{для } 40 \le j \le 59; \end{cases}$$

$$C_j = \begin{cases} [2^{30}\sqrt{2}] = 5a827999 & \text{для } 0 \le j \le 19, \\ [2^{30}\sqrt{3}] = 6ed9eba1 & \text{для } 20 \le j \le 39, \\ [2^{30}\sqrt{5}] = 8f1bbcdc & \text{для } 40 \le j \le 59, \\ [2^{30}\sqrt{10}] = ca62c1d6 & \text{для } 60 \le j \le 79. \end{cases}$$

($5a827999$ – восемь шестнадцатеричных цифр целой части числа $2^{30}\sqrt{2}$, другие константы получены аналогично).

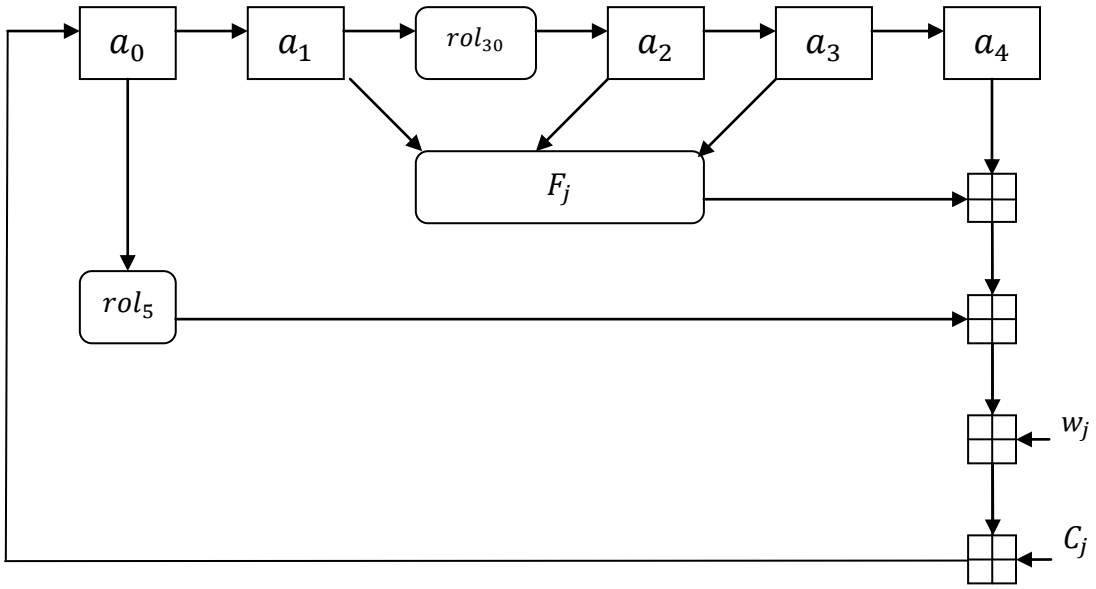


Рис. 9. *Элементарная операция SHA-1*

Таблица 9. *Псевдокод SHA-1*

$h := (0\text{x}67452301, 0\text{x}efcdab89, 0\text{x}98badcfe, 0\text{x}10325476, 0\text{x}c3d2e1f0);$
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ N\ \boldsymbol{do}\ \{$
    $a := h;$
    $\boldsymbol{for}\ j := 0\ \boldsymbol{to}\ 15\ \boldsymbol{do}\ w_j := M_{ij};$
    $\boldsymbol{for}\ j := 16\ \boldsymbol{to}\ 79\ \boldsymbol{do}\ w_j := rol_1(w_{j-3} \oplus w_{j-8} \oplus w_{j-14} \oplus w_{j-16});$
    $\boldsymbol{for}\ j := 0\ \boldsymbol{to}\ 79\ \boldsymbol{do}\ \{$
        $a_4 := rol_5(a_0) + F_j(a_1, a_2, a_3) + a_4 + w_j + C_j;$

$$(a_0, a_1, a_2, a_3, a_4) := (a_4, a_4, rol_{30}(a_1), a_2, a_3)$$

};

$h := (h_0 + a_0, h_1 + a_1, h_2 + a_2, h_3 + a_3, h_4 + a_4)$

}.

ГОСТ Р34.11-94

Хеш-функция, определенная в стандарте ГОСТ Р34.11-94, вычисляет 256-битовое хеш-значение сообщения произвольной длины. Данное хеш-значение используется для формирования и проверки электронной цифровой подписи в стандарте ГОСТ Р. 10-94. Исходное сообщение M разбивается на 256-битовые блоки M_1,…, M_N. Если последний блок не является полным, то он дополняется (слева) нулями до требуемого размера. В вычислении хеш-значения $h = H(M)$ используются два дополнительных 256-битовых блока: L, в котором записывается длина исходного сообщения (до дополнения), и Z, в котором формируется контрольная сумма.

Используемая при вычислении значения h функция сжатия f определяется следующим образом.

Пусть $c = 1^8 0^8 1^{16} 0^{24} 1^{16} 0^8 (0^8 1^8)^2 1^8 0^8 (0^8 1^8)^4 (1^8 0^8)^4$

– 256-битовый блок, где a^n – конкатенация n экземпляров набора a. В определении f используется функции S, P и T с 256-битовыми аргументами и значениями:

Функция S с аргументом $X = (X_4, X_3, X_2, X_1)$, где X_i – 64-битовые блоки, задана как

$$S(X) = (X_2 \oplus X_1, X_4, X_3, X_2).$$

Функция P с аргументом $B = (b_{32}, b_{31}, \dots, b_1)$, где b_i – байты, задана как

$$P(B) = (b_{\varphi(32)}, b_{\varphi(31)}, \dots, b_{\varphi(1)}),$$

где $\varphi(i) = ((i-1)\ div\ 4) + \ ((i-1)\ mod\ 4)\ 8 + 1$, $i = 1,2, \dots,32$.

Функция T с аргументом $W = (w_{16}, w_{15}, \dots, w_1)$, где w_i – 16-битовые блоки, задана как

$$T(W) = (v, w_{16}, w_{15}, \dots, w_2),$$

где $v = w_1 \oplus w_2 \oplus w_3 \oplus w_4 \oplus w_{13} \oplus w_{16}$.

Пусть $\mathcal{E}_K(Y)$ обозначает функцию зашифрования 64-битового блока Y под управлением 256-битового ключа K согласно ГОСТ 28147-89 (принципиально можно использовать любой другой алгоритм зашифрования с 64-битовым блоком и 256-битовым ключом). При таких обозначениях функция f определяется как (см. рис. 11*б*):

$f(m, h) \equiv \{$

$(U := m;\ V := h;\ (X_4, X_3, X_2, X_1) := m;$

$K_4 := P(U \oplus V);$

$U := S(U);\ V := S^2(V);$

$K_3 := P(U \oplus V);$

$U := S(U) \oplus c;\ V := S^2(V);$

$K_2 := P(U \oplus V);$

$U := S(U);\ V := S^2(V);$

$K_1 := P(U \oplus V);$

$W := (\mathcal{E}_{K_4}(X_4), \mathcal{E}_{K_3}(X_3), \mathcal{E}_{K_2}(X_2), \mathcal{E}_{K_1}(X_1));\ f := T^{61}(h \oplus T(m \oplus T^{12}(W)))$

}.

Здесь $S^2(V) = S(S(V))$, $T^n(x) = T(T^{n-1}(x))$.

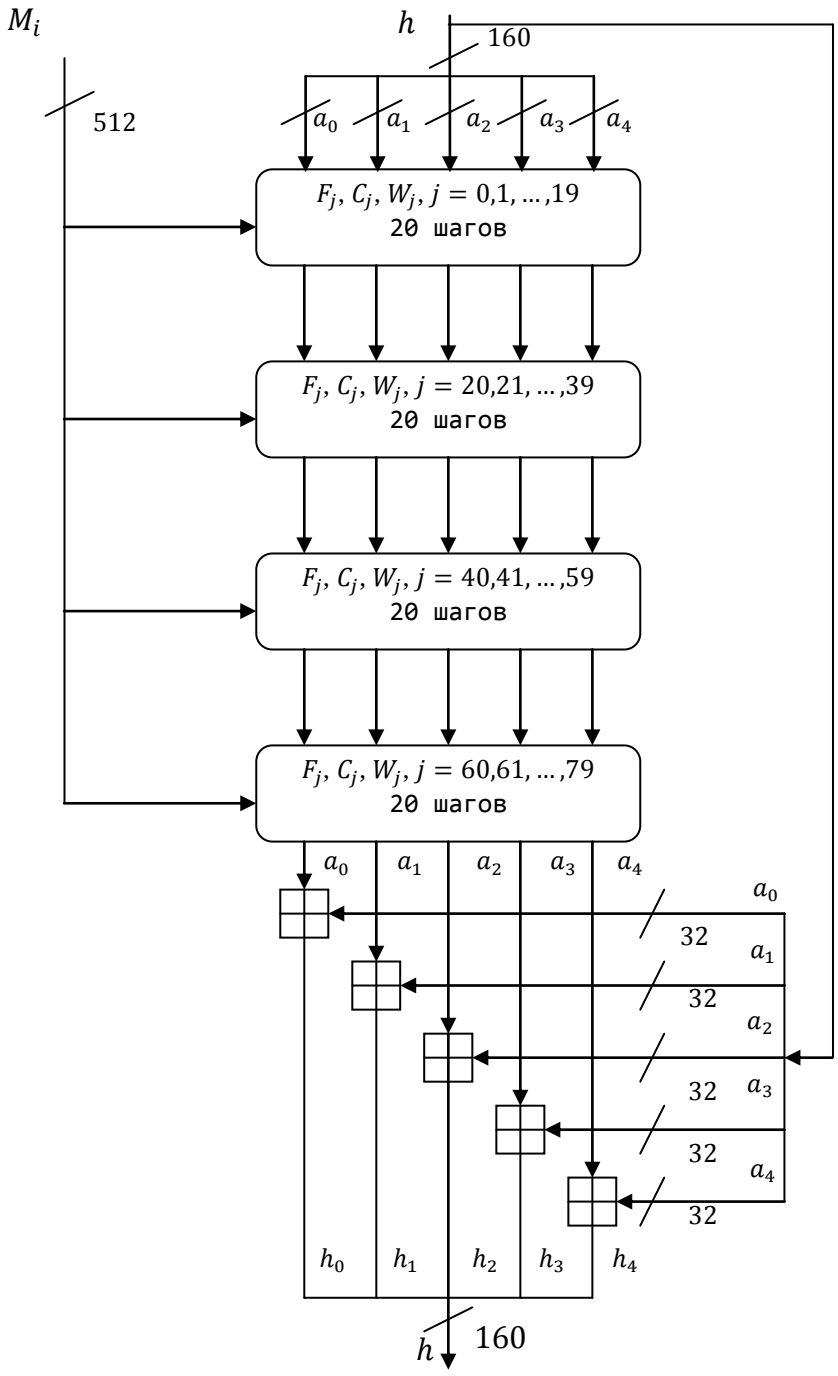
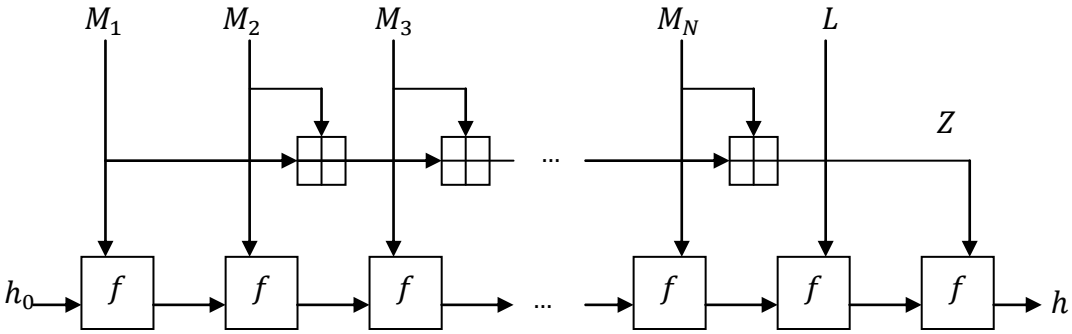


Рис.10. *Функция сжатия в* SHA-1 (*обработка одного* 512-*битового блока*)

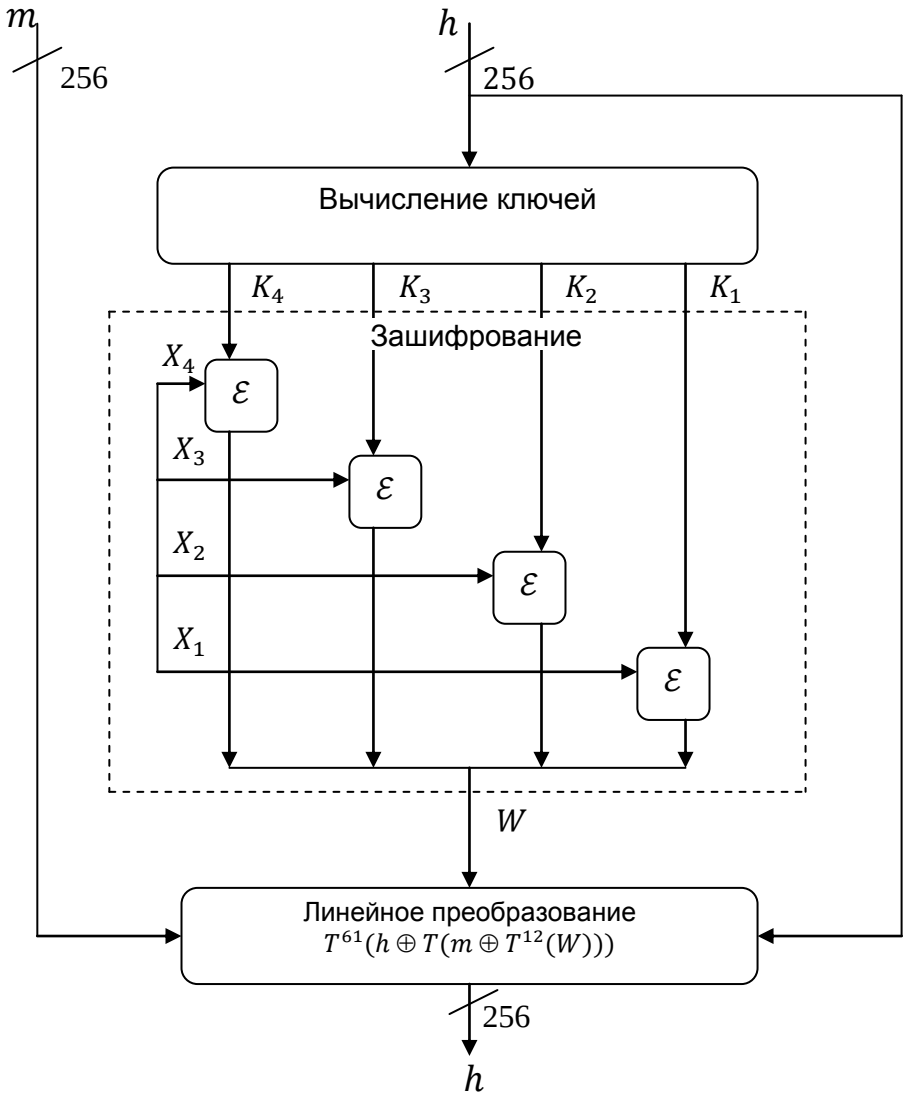
Хеш-значение $h = H(M)$ вычисляется по схеме (см.рис.$11a$):

$h := h_0;\ Z := 0;$
for $i := 1$ ***to*** N ***do*** {
$\quad h := f(M_i, h);\ Z := (Z + M_i)\ mod\ 2^{256}$
};
$h := f(L, h);$
$h := f(Z, h).$

Здесь h_0 – 256-битовый стартовый вектор хеширования, на его выбор в стандарте ограничений не накладывается.



(a)



$(б)$

Рис.11. *Функция хеширования в* ГОСТ Р34.11-94:
(a) *общая схема;* $(б)$ *функция сжатия* f

ГОСТ Р 34.11 – 2012

РОССИЙСКИЙ СТАНДАРТ ФУНКЦИ ХЕШИРОВАНИЯ

Определенная в стандарте функция хеширования используется при реализации электронной цифровой подписи на базе ассиметричного криптографического алгоритма по ГОСТ Р 34.10 –2012.Стандарт разработан взамен ГОСТ Р 34.11–94. Дата введения 01.01.2013. Основная часть стандарта дополнена одним приложением с контрольными примерами.

II.4.1 Обозначения

V^*	множество всех двоичных векторов (строк) конечной размерности, включая пустую строку;
| A |	размерность (число компонент) вектора A ∈ V^* (если A – пустая строка, то $\| A \| = 0$);
V_n	множество всех n-мерных двоичных векторов, где n − целое неотрицательное число; нумерация подвекторов и компонент вектора осуществляется ***справа налево, начиная с нуля***;
$\oplus$	операция покомпонентного сложения по модулю 2 двух двоичных векторов одинаковой размерности;
A||B	конкатенация векторов $A.B \in V^*$, т. е. вектор из $V_{\| A \| + \| B \|}$, в котором левый подвектор из $V_{\| A \|}$ совпадает с вектором A, а правый подвектор из $V_{\| B \|}$ совпадает с вектором B;
A^n	конкатенация n экземпляров вектора A;
$\mathbb{Z}_{2^n}$	кольцо вычетов по модулю 2^n;
$\boxplus$	операция сложения в кольце $\mathbb{Z}_{2^n}$;
$Vec_n \to V_n$	биективное отображение, сопоставляющее целому числу из $\mathbb{Z}_{2^n}$ его двоичное представление, т. е. для любого элемента $z = z_0 + 2z_1 + \cdots + 2^{n-1}z_{n-1}$ кольца $\mathbb{Z}_{2^n}$, где $z_j \in \{0,1\}$, $j = 0, \ldots, n-1$, выполнено равенство $Vec_n(z) = z_{n-1} \| \cdots \| z_1 \| z_0$;
$Int_n: V_n \to \mathbb{Z}_{2^n}$	отображение, обратное отображению Vec_n, т. е. $Int_n = Vec_n^{-1}$;
$MSB_n: V^* \to V_n$	отображение, ставящее в соответствие вектору $z_{k-1} \| \cdots \| z_1 \| z_0$, $k \geq n$, вектор $z_{k-1} \| \cdots \| z_{k-n+1} \| z_{k-n}$ (n старших разрядов исходного вектора);
$a := b$	операция присваивания переменной a значения b;
$\Phi \circ \Psi$	произведение отображений, при котором отображение Ψ действует ***первым***, т.е. $\Phi \circ \Psi(x) = \Phi(\Psi(x))$;
M	двоичный вектор, подлежащий хешированию, $M \in V^*$, $\| M \| < 2^{512}$;
$H: V^* \to V_n$	функция хеширования, отображающая вектор (сообщение) M в вектор (хеш-код) $H(M)$;
iv	инициализационный вектор функции хеширования, $iv \in V_{512}$.

II.4.2 Общие положения. Значения параметров

Данный стандарт определяет две функции хеширования $H: V^* \to V_n$ с длинами хеш-кода $n = 256$ бит и $n = 512$ бит.

Значение инициализационного вектора IV для функции хеширования с длиной хеш-кода 512 бит равно 0^{512}.

Значение инициализационного вектора iv для функции хеширования с длиной хеш-кода 256 бит равно $(00000001)^{64} = (0x01)^{64}$.

II.4.3. Нелинейное биективное преобразование множества двоичных векторов

Нелинейное биективное преобразование множества двоичных векторов V_8 задается подстановкой $\pi\colon \mathbb{Z}_{2^8} \to \mathbb{Z}_{2^8}$. Данная подстановка задана массивом

$$\pi = \big(\pi(0), \pi(1), \dots, \pi(255)\big) =$$

```
(252,238,221,17,207,110,49,22,251,196,250,218,35,197,4,77,233,119,240,219,147,46,153,
186,23,54,241,187,20,205,95,193,249,24,101,90,226,92,239,33,129,28,60,66,139,1,142,
79,5,132,2,174,227,106,143,160,6,11,237,152,127,212,211,31,235,52,44,81,234,200,72,
17,1,242,42,104,162,253,58,206,204,181,112,14,86,8,12,118,18,191,114,19,71,156,183,
93,135,21,161,150,41,16,123,154,199,243,145,120,111,157,158,178,177,50,117,25,61,
255,53,138,126,109,84,198,128,195,189,13,87,223,245,36,169,62,168,67,201,215,121,
214,246,124,34,185,3,224,15,236,222,122,148,176,188,220,232,40,80,78,51,10,74,167,
151,96,115,30,0,98,68,26,184,56,130,100,159,38,65,173,69,70,146,39,94,85,47,140,163,
165,125,105,213,149,59,7,88,179,64,134,172,29,247,48,55,107,228,136,217,231,137,225,
27,131,73,76,63,248,254,141,83,170,144,202,216,133,97,32,113,103,164,45,43,9,91,203,
155,37,208,190,229,108,82,89,166,116,210,230,244,180,192,209,102,175,194,57,75,99,
182).
```

II.4.4. Перестановка байт

Значения перестановки $\tau \in S_{64}$ записаны ниже в виде массива

$$\tau = \big(\tau(0), \tau(1), \dots, \tau(63)\big) =$$

```
(0,8,16,24,32,40,48,56,1,9,17,25,33,41,49,57,2,10,18,26,34,42,50,58,3,11,19,27,35,43,
51,59,4,12,20,28,36,44,52,60,5,13,21,29,37,45,53,61,6,14,22,30,38,46,54,62,7,15,23,
31,39,47,55,63).
```

II.4.5. Линейное преобразование множества двоичных векторов

Линейное преобразование l множества двоичных векторов V_{64} задается умножением ***справа*** на матрицу A над полем $GF(2)$, строки которой записаны ниже последовательно в шестнадцатеричном виде. Строка матрицы с номером j, $j = 0, \dots, 63$, записанная в виде $a_{j,15} \dots a_{j,0}$, где $a_{j,i} \in \mathbb{Z}_{16}$, $i = 0, \dots, 15$, есть $Vec_4(a_{j,15}) \parallel \cdots \parallel Vec_4(a_{j,0})$.

```
8e20faa72ba0b470 47107ddd9b505a38 ad08b0e0c3282d1c d8045870ef14980e
6c022c38f90a4c07 3601161cf205268d 1b8e0b0e798c13c8 83478b07b2468764
a011d380818e8f40 5086e740ce47c920 2843fd2067adea10 14aff010bdd87508
0ad97808d06cb404 05e23c0468365a02 8c711e02341b2d01 46b60f011a83988e
90dab52a387ae76f 486dd4151c3dfdb9 24b86a840e90f0d2 125c354207487869
092e94218d243cba 8a174a9ec8121e5d 4585254f64090fa0 accc9ca9328a8950
9d4df05d5f661451 c0a878a0a1330aa6 60543c50de970553 302a1e286fc58ca7
18150f14b9ec46dd 0c84890ad27623e0 0642ca05693b9f70 0321658cba93c138
86275df09ce8aaa8 439da0784e745554 afc0503c273aa42a d960281e9d1d5215
e230140fc0802984 71180a8960409a42 b60c05ca30204d21 5b068c651810a89e
456c34887a3805b9 ac361a443d1c8cd2 561b0d22900e4669 2b838811480723ba
9bcf4486248d9f5d c3e9224312c8c1a0 effa11af0964ee50 f97d86d98a327728
e4fa2054a80b329c 727d102a548b194e 39b008152acb8227 9258048415eb419d
492c024284fbaec0 aa16012142f35760 550b8e9e21f7a530 a48b474f9ef5dc18
70a6a56e2440598e 3853dc371220a247 1ca76e95091051ad 0edd37c48a08a6d8
07e095624504536c 8d70c431ac02a736 c83862965601dd1b 641c314b2b8ee083
```

Здесь в одной строке записано 4 строки матрицы A, при этом в строке с номером i, $i = 0, \dots, 15$ записаны строки матрицы A с номерами $4i + j$, $i = 0, \dots, 3$, в следующем порядке (слева направо) $4i + 0$, $4i + 1$, $4i + 2$,$4i + 3$. Результат умножения вектора $b = b_0 \dots b_{63} \in V_{64}$ на матрицу A есть вектор $c \in V_{64}$:

$$c = b_{63}\left(Vec_4(a_{0,15}) \parallel \cdots \parallel Vec_4(a_{0,0})\right) \oplus \dots \oplus b_0\left(Vec_4(a_{63,15}) \parallel \cdots \parallel Vec_4(a_{63,0})\right)$$

где

$$b_i\left(Vec_4(a_{63-i,15}) \parallel \cdots \parallel Vec_4(a_{63-i,0})\right) = \begin{cases} 0^{64}, & \text{если } b_i = 0, \\ Vec_4(a_{63-i,15}) \parallel \cdots \parallel Vec_4(a_{63-i,0}), & \text{если } b_i = 1 \end{cases}$$

для всех $i = 0, \dots, 63$.

II.4.6. Итерационные константы

Итерационные константы записаны в шестнадцатеричном виде. Значение константы,записанное в виде $a_{127} \dots a_0$, где $a_i \in \mathbb{Z}_{16}$, $i = 0, \dots, 127$, есть $Vec_4(a_{127}) \| \cdots \| Vec_4(a_0)$:

$C_{1(512)}$=
b1085bda 1ecadae9 ebcb2f81 c0657c1f 2f6a7643 2e45d016 714eb88d 7585c4fc
4b7ce091 92676901 a2422a08 a460d315 05767436 cc744d23 dd806559 f2a64507;
$C_{2(512)}$=
6fa3b58a a99d2f1a 4fe39d46 0f70b5d7 f3feea72 0a232b98 61d55e0f 16b50131
9ab5176b 12d69958 5cb561c2 db0aa7ca 55dda21b d7cbcd56 e6790470 21b19bb7;
$C_{3(512)}$=
f574dcac 2bce2fc7 0a39fc28 6a3d8435 06f15e5f 529c1f8b f2ea7514 b1297b7b
d3e20fe4 90359eb1 c1c93a37 6062db09 c2b6f443 867adb31 991e96f5 0aba0ab2;
$C_{4(512)}$=
ef1fdfb3 e81566d2 f948e1a0 5d71e4dd 488e857e 335c3c7d 9d721cad 685e353f
a9d72c82 ed03d675 d8b71333 935203be 3453eaa1 93e837f1 220cbebc 84e3d12e;
$C_{5(512)}$=
4bea6bac ad474799 9a3f410c 6ca92363 7f151c1f 1686104a 359e35d7 800fffbd
bfcd1747 253af5a3 dfff00b7 23271a16 7a56a27e a9ea63f5 601758fd 7c6cfe57;
$C_{6(512)}$=
ae4faeae 1d3ad3d9 6fa4c33b 7a3039c0 2d66c4f9 5142a46c 187f9ab4 9af08ec6
cffaa6b7 1c9ab7b4 0af21f66 c2bec6b6 bf71c572 36904f35 fa68407a 46647d6e;
$C_{7(512)}$=
f4c70e16 eeaac5ec 51ac86fe bf240954 399ec6c7 e6bf87c9 d3473e33 197a93c9
0992abc5 2d822c37 06476983 284a0504 3517454c a23c4af3 8886564d 3a14d493;
$C_{8(512)}$=
9b1f5b42 4d93c9a7 03e7aa02 0c6e4141 4eb7f871 9c36de1e 89b4443b 4ddbc49a
f4892bcb 929b0690 69d18d2b d1a5c42f 36acc235 5951a8d9 a47f0dd4 bf02e71e;
$C_{9(512)}$=
378f5a54 1631229b 944c9ad8 ec165fde 3a7d3a1b 25894224 3cd955b7 e00d0984
800a440b dbb2ceb1 7b2b8a9a a6079c54 0e38dc92 cb1f2a60 72614451 83235adb;
$C_{10(512)}$=
abbedea6 80056f52 382ae548 b2e4f3f3 8941e71c ff8a78db 1fffe18a 1b336103
9fe76702 af69334b 7a1e6c30 3b7652f4 3698fad1 153bb6c3 74b4c7fb 98459ced;
$C_{11(512)}$=
7bcd9ed0 efc889fb 3002c6cd 635afe94 d8fa6bbb ebab0761 20018021 14846679
8a1d71ef ea48b9ca efbacd1d 7d476e98 dea2594a c06fd85d 6bcaa4cd 81f32d1b;
$C_{12(512)}$=
378ee767 f11631ba d21380b0 0449b17a cda43c32 bcdf1d77 f82012d4 30219f9b
5d80ef9d 1891cc86 e71da4aa 88e12852 faf417d5 d9b21b99 48bc924a f11bd720.

II.4.7. Преобразования и функция сжатия

При вычислении хеш-кода $H(M)$ сообщения M используются следующие преобразования:

$X[k]: V_{512} \to V_{512}, X[k](a) = k \oplus a, k, a \in V_{512}$

$S: V_{512} \to c, S(a) = S(a_{63} \| \cdots \| a_0) = \pi(a_{63}) \| \cdots \| \pi(a_0),$

где $a = a_{63} \| \cdots \| a_0 \in \mathrm{V}_{512}, a_i \in V_8, i = 0, \dots, 63;$

$$P: V_{512} \to V_{512}, P(a) = P(a_{63} \| \cdots \| a_0) = a_{\tau(63)} \| \cdots \| a_{\tau(0)}, \qquad (5)$$

где $a = a_{63} \| \cdots \| a_0 \in \mathrm{V}_{512}, a_i \in V_8, i = 0, \dots, 63;$

$$L: V_{512} \to V_{512}, L(a) = L(a_7 \| \cdots \| a_0) = l(a_7) \| \cdots \| l(a_0), \qquad (6)$$

где $a = a_7 \| \cdots \| a_0 \in \mathrm{V}_{512}, a_i \in V_{64}, i = 0, \dots, 7.$

Значение хеш-кода сообщения $M \in V^*$ вычисляется с использованием итерационной процедуры. На каждой итерации вычисления хеш-кода используется функция сжатия:

$$g_N: V_{512} \times V_{512} \to V_{512}, N \in V_{512},$$

значение которой вычисляется по формуле:

$$g_N(h,m)=E(L\circ P\circ S(h\oplus N),m)\oplus h\oplus m,$$

где

$$E(K,m)=X[K_{13}]\circ L\circ P\circ S\circ X[K_{12}]\circ\ldots\circ L\circ P\circ S\circ X[K_2]\circ L\circ P\circ S\circ X[K_1](m).$$

Значения $K_i\in V_{512}$, $i=1,\ldots,13$, вычисляются следующим образом:

$$K_1=K=L\circ P\circ S(IV)=\begin{cases} L\circ P\circ S(0^{512}), & \text{если } |h|=512,\\ L\circ P\circ S((00000001)^{64}), & \text{если } |h|=256,\end{cases}$$

$$K_i=L\circ P\circ S(K_{i-1}\oplus C_{i-1}),\ i=2,\ldots,13.$$

Для краткости вместо $g_{0^{512}}$ будем обозначать через g_0.

II.4.8. Алгоритм вычисления хеш-функции *H*

Исходными данными для процедуры вычисления хеш-кода $H(M)$ является подлежащее хешированию сообщение $M\in V^*$ и $iv\in V_{512}$ – инициализационный вектор хеширования.

Алгоритм вычисления хеш-функции H состоит из следующих этапов:

Этап 1

Присвоить начальные значения текущих величин:

1.1. $h:=iv$;

1.2. $N:=0^{512}$;

1.3. $\sum:=0^{512}\in V_{512}$;

1.4. Перейти к этапу 2.

Этап 2

2.1. Если $|M|<512$, то к этапу 3.

В противном случае выполнить последовательность вычислений 2.2 – 2.7:

2.2. Вычислить подвектор $m\in V_{512}$ сообщения M: $M=M'\|m$.

2.3. $h:=g_N(h,m)$;

2.4. $N:=Vec_{512}(Int_{512}(N)\boxplus 512)$;

2.5. $\sum:=Vec_{512}(Int_{512}(\sum)\boxplus Int_{512}(m))$;

2.6. $M:=M'$;

2.7. Перейти к шагу 2.1.

Этап 3

3.1. $m:=0^{511-|M|}\|1\|$ M;

3.2. $h:=g_N(h,m)$;

3.3. $N:=Vec_{512}(Int_{512}(N)\boxplus|M|)$;

3.4. $\sum:=Vec_{512}(Int_{512}(\sum)\boxplus Int_{512}(m))$;

3.5. $h:=g_0(h,N)$;

3.6. Конец работы алгоритма: значение h является значением хеш-кода $H(M)$.

II.4.9. Приложение. Контрольные примеры к ГОСТ Р 34.10 –2012

Данное приложение, не являясь частью стандарта, содержит контрольные примеры к ГОСТ Р 34.10 –2012 для проверки правильности программной реализации.

Векторы из V^*записываются в шестнадцатеричном виде. Вектор A, записанный в виде $a_{n-1}\ldots a_0$ где $a_i\in\mathbb{Z}_{16}$, $i=0,\ldots,n-1$, есть $Vec_4(a_{n-1})||\ldots||Vec_4(a_0)$.

A.1 Пример 1. Необходимо вычислить хеш-код сообщения

M_1= 323130393837363534333231303938373635343332313039383736353433323
13039383736353433323130393837363534333231303938373635343332313039383736353433323130.

Функции хеширования с длиной хеш-кода 512 бит присваиваются значения:

$h:=IV:=0^{512}$;

$N:=0^{512}$;

$\sum:=0^{512}$.

Длина сообщения $|M_1|=504<512$**, поэтому происходит заполнение неполного блока:**

$m :=$
0132313039383736353433323130393837363534333231303938373635343332
3130393837363534333231303938373635343332313039383736353433323130.
Вычисляется значение
$K := L \circ P \circ S(h \oplus N) = L \circ P \circ S(0^{512})$.
После преобразования S:
$S(h \oplus N) =$
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc,
после преобразования P:
$P \circ S(h \oplus N) =$
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc,
после преобразования L:
$K := L \circ P \circ S(h \oplus N) =$
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574.
Затем выполняется преобразование $E(K, m)$:
Итерация 1. $K_1 =$
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574,
$X[K_1](m) =$
b2b1cd1ef7ec924286b7cf1cffe49c4c84b5c91afde694448abbcb18fbe09646
82b3c516f9e2904080b1cd1ef7ec924286b7cf1cffe49c4c84b5c91afde69444,
$S \circ X[K_1](m) =$
4645d95fc0beec2c432f8914b62d4efd3e5e37f14b097aead67de417c220b048
2492ac996667e0ebdf45d95fc0beec2c432f8914b62d4efd3e5e37f14b097aea,
$P \circ S \circ X[K_1](m) =$
46433ed624df433e452f5e7d92452f5ed98937e4acd989375f14f117995f14f1
c0b64bc266c0b64bbe2d092067be2d09ec4e7ab0e0ec4e7a2cfdea48eb2cfdea,
$L \circ P \circ S \circ X[K_1](m) =$
e60059d4d8e0758024c73f6f3183653f56579189602ae4c21e7953ebc0e212a0
ce78a8df475c2fd4fc43fc4b71c01e35be465fb20dad2cf690cdf65028121bb9,
$K_1 \oplus C_1 =$
028ba7f4d01e7f9d5848d3af0eb1d96b9ce98a6de0917562c2cd44a3bb516188
f8ff1cbf5cb3cc7511c1d6266ab47661b6f5881802a0e8576e0399773c72e073,
$S(K_1 \oplus C_1) =$
ddf644e6e15f5733bff249410445536f4e9bd69e200f3596b3d9ea737d70a1d7
d1b6143b9c9288357758f8ef78278aa155f4d717dda7cb12b211e87e7f19203d,
$P \circ S(K_1 \oplus C_1) =$
ddbf4eb3d17755b2f6f29bd9b658f4114449d6ea14f8d7e8e6419e733bef177e
e104207d9c78dd7f5f450f709227a719575335a1888acb20336f96d735a1123d,
$L \circ P \circ S(K_1 \oplus C_1) =$
d0b00807642fd78f13f2c3ebc774e80de0e902d23aef2ee9a73d010807dae9c1
88be14f0b2da27973569cd2ba051301036f728bd1d7eec33f4d18af70c46cf1e.
Итерация 2. $K_1 =$
d0b00807642fd78f13f2c3ebc774e80de0e902d23aef2ee9a73d010807dae9c1
88be14f0b2da27973569cd2ba051301036f728bd1d7eec33f4d18af70c46cf1e,
$L \circ P \circ S \circ X[K_2] \circ L \circ P \circ S \circ X[K_1](m) =$
18e77571e703d19548075c574ce5e50e0480c9c5b9f21d45611ab86cf32e352a
d91854ea7df8f863d46333673f62ff2d3efae1cd966f8e2a74ce49902799aad4.
Итерация 3. $K_3 =$
9d4475c7899f2d0bb0e8b7dac6ef6e6b44ecf66716d3a0f16681105e2d13712a
1a9387ecc257930e2d61014a1b5c9fc9e24e7d636eb1607e816dbaf927b8fca9,
$L \circ P \circ S \circ X[K_3] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
03dc0a9c64d42543ccdb62960d58c17e0b5b805d08a07406ece679d5f82b70fe
a22a7ea56e21814619e8749b308214575489d4d465539852cd4b0cd3829bef39.
Итерация 4. $K_4 =$
5c283daba5ec1f233b8c833c48e1c670dae2e40cc4c3219c73e58856bd96a72f
df9f8055ffe3c004c8cde3b8bf78f95f3370d0a3d6194ac5782487defd83ca0f,

$L \circ P \circ S \circ X[K_4] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
dbee312ea7301b0d6d13e43855e85db81608c780c43675bc93cfd82c1b4933b3
898a35b13e1878abe119e4dffb9de4889738ca74d064cd9eb732078c1fb25e04.
Итерация 5, $K_5 =$
109f33262731f9bd569cbc9317baa551d4d2964fa18d42c41fab4e37225292ec
2fd97d749378477904638846 9ae195c436fa7cba93f8239ceb5ffc818826470c,
$L \circ P \circ S \circ X[K_5] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
7fb3f15718d90e889f9fb7c38f527bec861c298afb9186934a93c9d96ade20df
109379bb9c1a1ffd0ad81fce7b45ccd54501e7d127e32874b5d7927b032de7a1.
Итерация 6. $K_6 =$
b32c9b02667911cf8f8a0877be9a170757e25026ccf41e67c6b5da70b1b87474
3e1135cfbefe244237555c676c153d99459bc382573aee2d85d30d99f286c5e7,
$L \circ P \circ S \circ X[K_6] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
95efa4e104f235824bae5030fe2d0f170a38de3c9b8fc6d8fa1a9adc2945c413
389a121501fa71a65067916b0c06f6b87ce18de1a2a98e0a64670985f47d73f1.
Итерация 7. $K_7 =$
8a13c1b195fd0886ac49989e7d84b08bc7b00e4f3f62765ece6050fcbabdc234
6c8207594714e8e9c9c7aad694edc922d6b01e17285eb7e61502e634559e32f1,
$L \circ P \circ S \circ X[K_7] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
7ea4385f7e5e40103bfb25c67e404c7524eec43e33b1d06557469c6049854304
32b43d941b77ffd476103338e9bd5145d9c1e18b1f262b58a81dceff6fc6535.
Итерация 8. $K_8 =$
52cec3b11448bb8617d0ddfbc926f2e88730cb9179d6decea5acbffd323ec376
4c47f7a9e13bb1db56c342034773023d617ff01cc546728e71dff8de5d128cac,
$L \circ P \circ S \circ X[K_8] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
b2426da0e58d5cfe898c36e797993f902531579d8ecc59f8dd8a60802241a456
1f290cf992eb398894424bf681636968c167e870967b1dd9047293331956daba.
Итерация 9. $K_9 =$
f38c5b7947e7736d502007a05ea64a4eb9c243cb82154aa138b963bbb7f28e74
d4d710445389671291d70103f48fd4d4c01fc415e3fb7dc61c6088afa1a1e735,
$L \circ P \circ S \circ X[K_9] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
5e0c9978670b25912dd1ede5bdd1cf18ed094d14c6d973b731d50570d0a9bca2
15415a15031fd20ddefb5bc61b96671d6902f49df4d2fd346ceebda9431cb075.
Итерация 10. $K_{10} =$
0740b3faa03ed39b257dd6e3db7c1bf56b6e18e40cdaabd30617cecbaddd618e
a5e61bb4654599581dd30c24c1ab877ad0687948286cfefaa7eef99f6068b315,
$L \circ P \circ S \circ X[K_{10}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
c1ddd840fe491393a5d460440e03bf451794e792c0c629e49ab0c1001782dd37
691cb6896f3e00b87f71d37a584c35b9cd8789fad55a46887e5b60e124b51a61.
Итерация 11. $K_{11} =$
185811cf3c2633aec8cfdfcae9dbb29347011bf92b95910a3ad71e5fca678e45
e374f088f2e5c29496e9695ce8957837107bb3aa56441af11a82164893313116,
$L \circ P \circ S \circ X[K_{11}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
3f75beaf2911c35d575088e30542b689c85b6b1607f8b800405941f5ab704284
7b9b08b58b4fbdd6154ed7b366fd3ee778ce647726ddb3c7d48c8ce8866a8435.
Итерация 12. $K_{12} =$
9d46bf66234a7ed06c3b2120d2a3f15e0fedd87189b75b3cd2f206906b5ee00d
c9a1eab800fb8cc5760b251f4db5cdef427052fa345613fd076451901279ee4c,
$L \circ P \circ S \circ X[K_{12}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
f35b0d889eadfcff73b6b17f33413a97417d96f0c4cc9d30cda8ebb7dcd5d1b0
61e620bac75b367370605f474ddc006003bec4c4d7ce59a73fbe6766934c55a2.
Итерация 13. $K_{13} =$
0f79104026b900d8d768b6e223484c9761e3c585b3a405a6d2d8565ada926c3f
7782ef127cd6b98290bf612558b4b60aa3cbc28fd94f95460d76b621cb45be70,
$X[K_{13}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
fc221dc8b814fc27a4de079d10097600209e5375776898961f70bded0647bd8f
1664cfa8bb8d8ff1e0df3e621568b66aa075064b0e81cce132c8d1475809ebd2.
Результат выполнения преобразования $g_N(h, m)$**:**
$h =$
fd102cf8812ccb1191ea34af21394f3817a86641445aa9a626488adb33738ebd
2754f6908cbbbac5d3ed0f522c50815c954135793fb1f5d905fee4736b3bdae2.

Изменяются значения переменных N и $\sum$:
$N =$
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000001f8,
$\sum =$
0132313039383736353433323130393837363534333231303938373635343332
3130393837363534333231303938373635343332313039383736353433323130.
Результат выполнения преобразования $g_0(h, N)$:
$h =$
5c881fd924695cf196c2e4fec20d14b642026f2a0b1716ebaabb7067d4d59752
3d2db69d6d3794622147a14f19a66e7f9037e1d662d34501a8901a5de7771d7c.
Результат выполнения преобразования $g_0(h, \Sigma)$:
$h =$
486f64c1917879417fef082b3381a4e211c324f074654c38823a7b76f830ad00
fa1fbae42b1285c0352f227524bc9ab16254288dd6863dccd5b9f54a1ad0541b.
Хеш-кодом сообщения M_1 является значение
$H(M_1)$ =
486f64c1917879417fef082b3381a4e211c324f074654c38823a7b76f830ad00
fa1fbae42b1285c0352f227524bc9ab16254288dd6863dccd5b9f54a1ad0541b.

А.1.2 ДЛЯ ФУНКЦИИ ХЕШИРОВАНИЯ С ДЛИНОЙ ХЕШ-КОДА 256 БИТ

присваиваются значения:

$h := IV := 0^{512}$;
$N := 0^{512}$;
$\sum := 0^{512}$.

Длина сообщения $|M_1| = 504 < 512$, поэтому происходит заполнение неполного блока:
$M :=$
0132313039383736353433323130393837363534333231303938373635343332
3130393837363534333231303938373635343332313039383736353433323130.
Вычисляется значение $K := L \circ P \circ S(h \oplus N) = L \circ P \circ S((00000001)^{64})$:
После преобразования S:
$S(h \oplus N) =$
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee,
после преобразования P:
$P \circ S(h \oplus N) =$
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee,
после преобразования L:
$K := L \circ P \circ S(h \oplus N) =$
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15.
Затем выполняется преобразование $E(K, m)$:
Итерация 1. $K_1 =$
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15,
$X[K_1](m) =$
22f7df708943682316f1dd72814b662d14f3db7483496e251afdd976854f6c27
12f5d778874d6a2110f7df708943682316f1dd72814b662d14f3db7483496e25,
$S \circ X[K_1](m) =$
65c061327951f35a99a6d819f5a29a0193d290ffa92ab25cf14b538aa8cc9d21
f0f4fe6dc93a7818e9c061327951f35a99a6d819f5a29a0193d290ffa92ab25c,
$P \circ S \circ X[K_1](m) =$
659993f1f0e99993c0a6d24bf4c0a6d261d89053fe61d8903219ff8a6d3219ff
79f5a9a8c979f5a951a22acc3a51a22af39ab29d78f39ab25a015c21185a015c,
$L \circ P \circ S \circ X[K_1](m) =$
e549368917a0a2611d5e08c9c2fd5b3c563f18c0f68c410d84ae9d5fbdfb9340
55650121b7aa6d7b3e7d09d46ac4358adaa6ae44fa3b0402c4166d2c3eb2ef02,
$K_1 \oplus C_1 =$
92cdb59aaeb185fcc80ec1c1701e230a0caf98039e3e8f03528b56cdc5fe9be9
68b90ed1221c36148187c448141b8c0026b39a767c0f1236fe458b1942dd1a12,

$S(K_1 \oplus C_1) =$
ecd95e282645a83930045858325f5afa2341dc110ad303110ef676d9ac63509b
f3a3041b65148f93f5c986f293bb7cfcef92288ac34df08f63c8f6362cd8f1f0,
$P \circ S(K_1 \oplus C_1) =$
ec30230ef3f5ef63d90441f6a3c992c85e58dc76048628f6285811d91bf28a36
26320aac6593c32c455fd36314bb4dd8a85a03508f7cf0f139fa119b93fc8ff0,
$L \circ P \circ S(K_1 \oplus C_1) =$
18ee8f3176b2ebea3bd6cb8233694cea349769df88be26bf451cfab6a904a549
da22de93a66a66b19c7e6b5eea633511e611d68c8401bfcd0c7d0cc39d4a5eb9.
Итерация 2. $K_2 =$
18ee8f3176b2ebea3bd6cb8233694cea349769df88be26bf451cfab6a904a549
da22de93a66a66b19c7e6b5eea633511e611d68c8401bfcd0c7d0cc39d4a5eb9,
$L \circ P \circ S \circ X[K_2] \circ L \circ P \circ S \circ X[K_1](m) =$
c502dab7e79eb94013fcd1ba64def3b916f18b63855d43d22b77fca1452f9866
c2b45089c62e9d82edf1ef45230db9a23c9e1c521113376628a5f6a5dbc041b2.
Итерация 3. $K_3 =$
aaa4cf31a265959157aec8ce91e7fd46bf27dee21164c5e3940bba1a519e9d1f
ce0913f1253e7757915000cd674be12cc7f68e73ba26fb00fd74af4101805f2d,
$L \circ P \circ S \circ X[K_3] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
8e5a4fe41fc790af29944f027aa2f10105d65cf60a66e442832bb9ab5020dc54
772e36b03d4b9aa471037212cde93375226552392ef4d83010a007e1117a07b5.
Итерация 4. $K_4 =$
61fe0a65cc177af50235e2afadded326a5329a2236747bf8a54228aeca9c4585
cd801ea9dd743a0d98d01ef0602b0e332067fb5ddd6ac1568200311920839286,
$L \circ P \circ S \circ X[K_4] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
dee0b40df69997afef726f03bdc13cb6ba9287698201296f2fd8284f06d33ea4
a850a0ff48026dd47c1e88ec813ed2eb1186059d842d8d17f0bfa259e56655b1.
Итерация 5. $K_5 =$
9983685f4fd3636f1fd5abb75fbf26a8e2934314aa2ecb3ee4693c86c06c7d4e
169bd540af75e1610a546acd63d960bad595394cc199bf6999a5d5309fe73d5a,
$L \circ P \circ S \circ X[K_5] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
675ea894d326432e1af7b201bc369f8ab021f6fa58da09678ffc08ef30db43a3
7f1f7347cb77da0f6ba30c85848896c3bac240ab14144283518b89a33d0caf07.
Итерация 6. $K_6 =$
f05772ae2ce7f025156c9a7fbcc6b8fdf1e735d613946e32922994e52820ffea
62615d907eb0551ad170990a86602088af98c83c22cdb0e2be297c13c0f7a156,
$L \circ P \circ S \circ X[K_6] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
1bc204bf9506ee9b86bbcf82d254a112aea6910b6db3805e399cb718d1b33199
64459516967cee4e648e8cfbf81f56dc8da6811c469091be5123e6a1d5e28c73.
Итерация 7. $K_7 =$
5ad144c362546e4e46b3e7688829fbb77453e9c3211974330b2b8d0e6be2b5ac
c89eb6b35167f159b7b005a43e5959a651a9b18cfc8e4098fcf03d9b81cfbb8d,
$L \circ P \circ S \circ X[K_7] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
f30d791ed78bdee819022a3d78182242124efcdd54e203f23fb2dc7f94338ff9
55a5afc15ffef03165263c4fdb36933aa982016471fbac9419f892551e9e568b.
Итерация 8. $K_8 =$
6a6cec9a1ba20a8db64fa840b934352b518c638ed530122a83332fe0b8efdac9
018287e5a9f509c78d6c746adcd5426fb0a0ad5790dfb73fc1f191a539016daa,
$L \circ P \circ S \circ X[K_8] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
1fc20f1e91a1801a4293d3f3aa9e91560fcc3810bb15f3ee9741c9b87452519f
67cb9145519884a24de6db736a5cb1430da7458e5e51b80be5204ba5b2600177.
Итерация 9.$K_9 =$
99217036737aa9b38a8d6643f705bd51f351531f948f0fc5e35fa35fee9dd8bd
bb4c9d580a224e9cd82e0e2069fc49ed367d5f94374435382b8fb6a8f5dd0409,
$L \circ P \circ S \circ X[K_9] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
1a52f09d1e81515a36171e0b1a2809c50359bed90f2e78cbd89b7d4afa6d0466
55c96bdae6ee97055cc7e857267c2ccf28c8f5dd95ed58a9a68c12663bb28967.
Итерация 10. $K_{10} =$
906763c0fc89fa1ae69288d8ec9e9dda9a7630e8bfd6c3fed703c35d2e62aeaf
f0b35d80a7317a7f76f83022f2526791ca8fdf678fcb337bd74fe5393ccb05d2,

$L \circ P \circ S \circ X[K_{10}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
764043744a0a93687e65aba8cfc25ec8714fb8e1bdc9ae2271e7205eaaa577c1
b3b83e7325e50a19bd2d56b061b5de39235c9c9fd95e071a1a291a5f24e8c774.
Итерация 11. $K_{11} =$
88ce996c63618e6404a5c8e03ee433854e2ae3eee68991bbbff3c29d38dadb6e
d6a1dae9a6dc6ddf52ce34af272f96d3159c8c624c3fe6e13d695c0bfc89add5,
$L \circ P \circ S \circ X[K_{11}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
9b1ce8ff26b445cb288c0aeccf84658eea91dbdf14828bf70110a5c9bd146cd9
646350cff4e90e7b63c5cc325e9b441081935f282d4648d9584f71860538f03b.
Итерация 12. $K_{12} =$
3e0a281ea9bd46063eec550100576f3a506aa168cf82915776b978fccaa32f38
b55f30c79982ca45628e8365d8798477e75a49c68199112a1d7b5a0f7655f2db,
$L \circ P \circ S \circ X[K_{12}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
133aeecede251eb81914b8ba48dcbc0b8a6fc63a292cc49043c3d3346b3f0829
a9cb71ecff25ed2a91bdcf8f649907c110cb76ff2e43100cdd4ba8a147a572f5.
Итерация 13. $K_{13} =$
f0b273409eb31aebe432fbae1867212262c848422b6a92f93f6cbab54ed18b83
14b21cffc51e3fa319ff433e76ef6adb0ef9f5e03c907fa1fcf9eca06500bf03,
$L \circ P \circ S \circ X[K_{13}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
e3889d8e40960453fd26431450bb9d29e8a78e78024656697caf698125ee83aa
bd796d133a3bd28988428cb112766d1a1e32831f12d36fad21b2440122a5cdf6.
Результат выполнения преобразования $g_N(h, m)$**:**
$h =$
e3bbadbf78af3264c9137127608aa510de90ba4d3075665844965fb611dbb199
8d48552a0c0ce6bcba71bc802a4f5b2d2a07b12c22e25794178570341096fdc7.
Изменяются значения переменных N **и** $\sum$:
$N =$
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000001f8,
$\sum =$
0132313039383736353433323130393837363534333231303938373635343332
3130393837363534333231303938373635343332313039383736353433323130.
Результат выполнения преобразования $g_0(h, m)$*:*
$h =$
70f22bada4cfe18a6a56ec4b3f328cd40db8e1bf8a9d5f711d5efab11191279d
715aab7648d07eddbf87dc79c80516e6ffcbcf5678b0ac29ea00fa85c8173cc6.
Результат выполнения преобразования $g_0(h, \sum)$**:**
$h =$
00557be5e584fd52a449b16b0251d05d27f94ab76cbaa6da890b59d8ef1e159d
2088e482e2acf564e0e9795a51e4dd261f3f667985a2fcc40ac8631faca1709a.
Хеш-кодом сообщения M_1 **является значение:**
$H(M_1) =$
00557be5e584fd52a449b16b0251d05d27f94ab76cbaa6da890b59d8ef1e159d.

A.2 ПРИМЕР 2. ПУСТЬ НЕОБХОДИМО ВЫЧИСЛИТЬ ХЕШ-КОД СООБЩЕНИЯ

$M_2 =$
fbe2e5f0eee3c820fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0
eeec20f120faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ce8f0f2e5e220e5d1.

A.2.1 Для функции хеширования с длиной хеш-кода 512 бит присваиваются значения:

$h := IV := 0^{512}$;
$N := 0^{512}$;
$\sum := 0^{512}$.

Длина сообщения $|M_2| = 576 > 512$**, поэтому сначала преобразуется часть сообщения**
$m =$
fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0eeec20f1
20faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ce8f0f2e5e220e5d1.
$K := L \circ P \circ S(h \oplus N) = L \circ P \circ S(0^{512}),$
Вычисляется значение K**.**
После преобразования S**:**

$S(h \oplus N) =$
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc,
после преобразования P**:**
$P \circ S(h \oplus N) =$
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc
fcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfcfc,
после преобразования L**:**
$K := L \circ P \circ S(h \oplus N) =$
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574.
Затем выполняется преобразование $E(K, m)$**:**
Итерация 1. $K_1 =$
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574
b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574b383fc2eced4a574,
$X[K_1](m) =$
486906c521f45a8f43621cde3bf44599936b10ce2531558642a303de20388585
93790ed02b3685585b750fc32cf44d925d6214de3c0585585b730ecb2cf440a5,
$S \circ X[K_1](m) =$
f29131ac18e613035196148598e6c8e8de6fe9e75c840c432c731185f906a8a8
de5404e1428fa8bf47354d408be63aecb79693857f6ea8bf473d04e48be6eb00,
$P \circ S \circ X[K_1](m) =$
f251de2cde47b74791966f735435963d3114e911044d9304ac85e785e14085e4
18985cf9428b7f8be6e684068fe66ee613c80ca8a83aa8eb03e843a8bfecbf00,
$L \circ P \circ S \circ X[K_1](m) =$
909aa733e1f52321a2fe35bfb8f67e92fbc70ef544709d5739d8faaca4acf126
e83e273745c25b7b8f4a83a7436f6353753cbbbe492262cd3a868eace0104af1,
$K_1 \oplus C_1 =$
028ba7f4d01e7f9d5848d3af0eb1d96b9ce98a6de0917562c2cd44a3bb516188
f8ff1cbf5cb3cc7511c1d6266ab47661b6f5881802a0e8576e0399773c72e073,
$S(K_1 \oplus C_1) =$
ddf644e6e15f5733bff249410445536f4e9bd69e200f3596b3d9ea737d70a1d7
d1b6143b9c9288357758f8ef78278aa155f4d717dda7cb12b211e87e7f19203d,
$P \circ S(K_1 \oplus C_1) =$
ddbf4eb3d17755b2f6f29bd9b658f4114449d6ea14f8d7e8e6419e733bef177e
e104207d9c78dd7f5f450f709227a719575335a1888acb20336f96d735a1123d,
$L \circ P \circ S(K_1 \oplus C_1) =$
d0b00807642fd78f13f2c3ebc774e80de0e902d23aef2ee9a73d010807dae9c1
88be14f0b2da27973569cd2ba051301036f728bd1d7eec33f4d18af70c46cf1e.
Итерация 2. $K_2 =$
d0b00807642fd78f13f2c3ebc774e80de0e902d23aef2ee9a73d010807dae9c1
88be14f0b2da27973569cd2ba051301036f728bd1d7eec33f4d18af70c46cf1e,
$L \circ P \circ S \circ X[K_2] \circ L \circ P \circ S \circ X[K_1](m) =$
301aadd761d13df0b473055b14a2f74a45f408022aecadd4d5f19cab8228883a
021ac0b62600a495950c628354ffce1161c68b7be7e0c58af090ce6b45e49f16.
Итерация 3. $K_3 =$
9d4475c7899f2d0bb0e8b7dac6ef6e6b44ecf66716d3a0f16681105e2d13712a
1a9387ecc257930e2d61014a1b5c9fc9e24e7d636eb1607e816dbaf927b8fca9,
$L \circ P \circ S \circ X[K_3] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
9b83492b9860a93cbca1c0d8e0ce59db04e10500a6ac85d4103304974e78d32
259ceff03fbb353147a9c948786582df78a34c9bde3f72b3ca41b9179c2cceef3.
Итерация 4, $K_4 =$
5c283daba5ec1f233b8c833c48e1c670dae2e40cc4c3219c73e58856bd96a72f
df9f8055ffe3c004c8cde3b8bf78f95f3370d0a3d6194ac5782487defd83ca0f,
$L \circ P \circ S \circ X[K_4] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
e638e0a1677cdea107ec3402f70698a4038450dab44ac7a447
e10155aa33ef1bdaf8f49da7b66f3e05815045fbd39c991cb0dc536e09505fd62d3c2cd00b0f57.
Итерация 5. $K_5 =$
109f33262731f9bd569cbc9317baa551d4d2964fa18d42c41fab4e37225292ec
2fd97d7493784779046388469ae195c436fa7cba93f8239ceb5ffc818826470c,

$L \circ P \circ S \circ X[K_5] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
1c7c8e19b2bf443eb3adc0c787a52a173821a97bc5a8efea58fb8b27861829f6
dd5ff9c97865e08c1ac66f47392b578e21266e323a0aacedeec3ef0314f517c6.
Итерация 6. $K_6 =$
b32c9b02667911cf8f8a0877be9a170757e25026ccf41e67c6b5da70b1b87474
3e1135cfbefe244237555c676c153d99459bc382573aee2d85d30d99f286c5e7,
$L \circ P \circ S \circ X[K_6] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
48fecfc5b3eb77998fb39bfcccd128cd42fccb714221be1e675a1c6fdde7e311
98b318622412af7e999a3eff45e6d61609a7f2ae5c2ff1ab7ff3b37be7011ba2.
Итерация 7. $K_7 =$
8a13c1b195fd0886ac49989e7d84b08bc7b00e4f3f62765ece6050fcbabdc234
6c8207594714e8e9c9c7aad694edc922d6b01e17285eb7e61502e634559e32f1,
$L \circ P \circ S \circ X[K_7] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
a48f8d781c2c5be417ae644cc2e15a9f01fcead3232e5bd53f18a5ab875cce1b
8a1a400cf48521c7ce27fb1e94452fb54de23118f53b364ee633170a62f5a8a9.
Итерация 8. $K_8 =$
52cec3b11448bb8617d0ddfbc926f2e88730cb9179d6decea5acbffd323ec376
4c47f7a9e13bb1db56c342034773023d617ff01cc546728e71dff8de5d128cac,
$L \circ P \circ S \circ X[K_8] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
e8a31b2e34bd2ae21b0ecf29cc4c37c75c4d11d9b82852517515c23e81e906a4
51b72779c3087141f1a15ab57f96d7da6c7ee38ed25befbdef631216356ff59c.
Итерация 9. $K_9 =$
f38c5b7947e7736d502007a05ea64a4eb9c243cb82154aa138b963bbb7f28e74
d4d710445389671291d70103f48fd4d4c01fc415e3fb7dc61c6088afa1a1e735,
$L \circ P \circ S \circ X[K_9] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
34392ed32ea3756e32979cb0a2247c3918e0b38d6455ca88183356bf8e5877e5
5d542278a696523a8036af0f1c2902e9cbc585de803ee4d26649c9e1f00bda31.
Итерация 10. $K_{10} =$
0740b3faa03ed39b257dd6e3db7c1bf56b6e18e40cdaabd30617cecbaddd618e
a5e61bb4654599581dd30c24c1ab877ad0687948286cfefaa7eef99f6068b315,
$L \circ P \circ S \circ X[K_{10}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
6a82436950177fea74cce6d507a5a64e54e8a3181458e3bdfbdbc6180c9787de7
ccb676dd809e7cb1eb2c9ebd016561570801a4e9ce17a438b85212f4409bb5e.
Итерация 11. $K_{11} =$
185811cf3c2633aec8cfdfcae9dbb29347011bf92b95910a3ad71e5fca678e45
e374f088f2e5c29496e9695ce8957837107bb3aa56441af11a82164893313116,
$L \circ P \circ S \circ X[K_{11}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
7b97603135e2842189b0c9667596e96bd70472ccbc73ae89da7d1599c72860c2
85f5771088f1fb0f943d949f22f1413c991eafb51ab8e5ad8644770037765aec.
Итерация 12. $K_{12} =$
9d46bf66234a7ed06c3b2120d2a3f15e0fedd87189b75b3cd2f206906b5ee00d
c9a1eab800fb8cc5760b251f4db5cdef427052fa345613fd076451901279ee4c,
$L \circ P \circ S \circ X[K_{12}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
39ec8a88db635b46c4321adf41fd9527a39a67f6d7510db5044f05efaf721db5
cf976a726ef33dc4dfcda94033e741a463770861a5b25fefcb07281eed629c0e.
Итерация 13. $K_{13} =$
0f79104026b900d8d768b6e223484c9761e3c585b3a405a6d2d8565ada926c3f
7782ef127cd6b98290bf612558b4b60aa3cbc28fd94f95460d76b621cb45be70,
$L \circ P \circ S \circ X[K_{13}] \circ \ldots \circ L \circ P \circ \ S \circ X[K_1](m) =$
36959ac8fdda5b9e135aac3d62b5d9b0c279a27364f50813d69753b575e0718a
b8158560122584464f72c8656b53f7aec0bccaee7cfdcaa9c6719e3f2627227e.
Результат выполнения преобразования $g_N(h, m)$**:**
$h =$
cd7f602312faa465e3bb4ccd9795395de2914e938f10f8e127b7ac459b0c517b
98ef779ef7c7a46aa7843b8889731f482e5d221e8e2cea852e816cdac407c7af.
Изменяются значения переменных N **и** Σ**:**
$N =$
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000200,

Σ=
fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0eeec20f1
20faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ce8f0f2e5e220e5d1.
Длина оставшейся части сообщения меньше 512, поэтому происходит заполнение неполного блока.
$m :=$
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000001fbe2e5f0eee3c820.
Результат выполнения преобразования $g_N(h, m)$:
$h =$
c544ae6efdf14404f089c72d5faf8dc6aca1db5e28577fc07818095f1df70661
e8b84d0706811cf92dffb8f96e61493dc382795c6ed7a17b64685902cbdc878e.
Изменяются значения переменных N и Σ:
$N =$
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000240,
$\Sigma =$
fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0eeec20f1
20faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ee4d3d8d6d104adf1.
Результат выполнения преобразования $g_0(h, m)$:
$h =$
4deb6649ffa5caf4163d9d3f9967fbbd6eb3da68f916b6a09f41f2518b81292b
703dc5d74e1ace5bcd3458af43bb456e837326088f2b5df14bf83997a0b1ad8d.
Результат выполнения преобразования $g_0(h, \Sigma)$:
$h =$
28fbc9bada033b1460642bdcddb90c3fb3e56c497ccd0f62b8a2ad4935e85f03
7613966de4ee00531ae60f3b5a47f8dae06915d5f2f194996fcabf2622e6881e.
Хеш-кодом сообщения M_2 является значение:
$H(M_2) =$
28fbc9bada033b1460642bdcddb90c3fb3e56c497ccd0f62b8a2ad4935e85f03
7613966de4ee00531ae60f3b5a47f8dae06915d5f2f194996fcabf2622e6881e.

А.2.2 ДЛЯ ФУНКЦИИ ХЕШИРОВАНИЯ С ДЛИНОЙ ХЕШ-КОДА 256 БИТ

Присваиваются значения:

$h := IV = (00000001)^{64}$

$N := 0^{512}$;

$\sum := 0^{512}$.

Длина сообщения $|M_2| = 576 > 512$, поэтому сначала преобразуется часть сообщения
$m :=$
fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0eeec20f1
20faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ce8f0f2e5e220e5d1.
Вычисляется значение $K := L \circ P \circ S(h \oplus N) = L \circ P \circ S(0^{64})$.
После преобразования S:
$S(h \oplus N) =$
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee,
после преобразования P:
$P \circ S(h \oplus N) =$
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee,
после преобразования L:
$K := L \circ P \circ S(h \oplus N) =$
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15.
Затем выполняется преобразование $\mathbf{E}(\mathbf{K}, \mathbf{m})$:
Итерация 1. $K_1 =$
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15
23c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f1523c5ee40b07b5f15,
$X[K_1](m) =$
d82f14ab5f5ba0eed3240eb0455bbff8032d02a05b9eafe7d2e511b05e977fe4
033f1cbe55997f39cb331dad525bb7f3cd2406b042aa7f39cb351ca5525bbac4,

$S \circ X[K_1](m)$=
8d4f93828747a76c49e204adc8473bd11101dda7470a415b832b77ad5dbc572d
111f14950ce8570be4aecd9f0e472fd2d9e231ad2c38570be46a14000e47a586,
$P \circ S \circ X[K_1](m)$=
8d49118311e4d9e44fe2012b1faee26a9304dd7714cd311482ada7ad959fad00
87c8475d0c0e2c0e47470abce8473847a73b4157572f57a56cd15b2d0bd20b86,
$L \circ P \circ S \circ X[K_1](m)$=
a3a72a2e0fb5e6f812681222fec037b0db972086a395a387a6084508cae13093
aa71d352dcbce288e9a39718a727f6fd4c5da5d0bc10fac3707ccd127fe45475,
$K_1 \oplus C_1 =$
92cdb59aaeb185fcc80ec1c1701e230a0caf98039e3e8f03528b56cdc5fe9be9
68b90ed1221c36148187c448141b8c0026b39a767c0f1236fe458b1942dd1a12,
$S(K_1 \oplus C_1)$=
ecd95e282645a83930045858325f5afa2341dc110ad303110ef676d9ac63509b
f3a3041b65148f93f5c986f293bb7cfcef92288ac34df08f63c8f6362cd8f1f0,
$P \circ S(K_1 \oplus C_1)$=
ec30230ef3f5ef63d90441f6a3c992c85e58dc76048628f6285811d91bf28a36
26320aac6593c32c455fd36314bb4dd8a85a03508f7cf0f139fa119b93fc8ff0,
$L \circ P \circ S(K_1 \oplus C_1)$ =
18ee8f3176b2ebea3bd6cb8233694cea349769df88be26bf451cfab6a904a549
da22de93a66a66b19c7e6b5eea633511e611d68c8401bfcd0c7d0cc39d4a5eb9.
Итерация 2. $K_2 =$
18ee8f3176b2ebea3bd6cb8233694cea349769df88be26bf451cfab6a904a549
da22de93a66a66b19c7e6b5eea633511e611d68c8401bfcd0c7d0cc39d4a5eb9,
$L \circ P \circ S \circ X[K_2] \circ L \circ P \circ S \circ X[K_1](m) =$
9f50697b1d9ce23680db1f4d35629778864c55780727aa79eb7bb7d648829cba
8674afdac5c62ca352d77556145ca7bc758679fbe1fbd32313ca8268a4a603f1.
Итерация 3. $K_3 =$
aaa4cf31a265959157aec8ce91e7fd46bf27dee21164c5e3940bba1a519e9d1f
ce0913f1253e7757915000cd674be12cc7f68e73ba26fb00fd74af4101805f2d,
$L \circ P \circ S \circ X[K_3] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
4183027975b257e9bc239b75c977ecc52ddad82c091e694243c9143a945b4d85
3116eae14fd81b14bb47f2c06fd283cb6c5e61924edfaf971b78d771858d5310.
Итерация 4. $K_4 =$
61fe0a65cc177af50235e2afadded326a5329a2236747bf8a54228aeca9c4585
cd801ea9dd743a0d98d01ef0602b0e332067fb5ddd6ac1568200311920839286,
$L \circ P \circ S \circ X[K_4] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
0368c884fcee489207b5b97a133ce39a1ebfe5a3ae3cccb3241de1e7ad72857e
76811d324f01fd7a75e0b669e8a22a4d056ce6af3e876453a9c3c47c767e5712.
Итерация 5. $K_5 =$
9983685f4fd3636f1fd5abb75fbf26a8e2934314aa2ecb3ee4693c86c06c7d4e
169bd540af75e1610a546acd63d960bad595394cc199bf6999a5d5309fe73d5a,
$L \circ P \circ S \circ X[K_5] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
c31433ceb8061e46440144e65553976512e5a9806ac9a2c771d5932d5f6508c5
b78e406c4efab98ac5529be0021b4d58fa26f01621eb10b43de4c4c47b63f615.
Итерация 6. $K_6 =$
f05772ae2ce7f025156c9a7fbcc6b8fdf1e735d613946e32922994e52820ffea
62615d907eb0551ad170990a86602088af98c83c22cdb0e2be297c13c0f7a156,
$L \circ P \circ S \circ X[K_6] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
5d0ae97f252ad04534503fe5f52e9bd07f483ee3b3d206beadc6e736c6e754bb
713f97ea7339927893eacf2b474a482cadd9ac2e58f09bcb440cf36c2d14a9b6.
Итерация 7. $K_7 =$
5ad144c362546e4e46b3e7688829fbb77453e9c3211974330b2b8d0e6be2b5ac
c89eb6b35167f159b7b005a43e5959a651a9b18cfc8e4098fcf03d9b81cfbb8d,
$L \circ P \circ S \circ X[K_7] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
a59aa21e6ad3e330deedb9ab9912205c355b1c479fdfd89a7696d7de66fbf7d3
cec25879f7f1a8cca4c793d5f2888407aecb188bda375eae586a8cfd0245c317.

Итерация 8. $K_8 =$
6a6cec9a1ba20a8db64fa840b934352b518c638ed530122a83332fe0b8efdac9
018287e5a9f509c78d6c746adcd5426fb0a0ad5790dfb73fc1f191a539016daa,
$L \circ P \circ S \circ X[K_8] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
9903145a39d5a8c83d28f70fa1fbd88f31b82dc7cfe17b54b50e276cb2c4ac68
2b4434163f214cf7ce6164a75731bcea5819e6a6a6fea99da9222951d2a28e01.
Итерация 9. $K_9 =$
99217036737aa9b38a8d6643f705bd51f351531f948f0fc5e35fa35fee9dd8bd
bb4c9d580a224e9cd82e0e2069fc49ed367d5f94374435382b8fb6a8f5dd0409,
$L \circ P \circ S \circ X[K_9] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
330e6cb1d04961826aa263f2328f15b4f3370175a6a9fd6505b286efed2d8505
f71823337ef71513e57a700eb1672a685578e45dad298ee2223d4cb3fda8262f.
Итерация 10. $K_{10} =$
906763c0fc89fa1ae69288d8ec9e9dda9a7630e8bfd6c3fed703c35d2e62aeaf
f0b35d80a7317a7f76f83022f2526791ca8fdf678fcb337bd74fe5393ccb05d2,
$L \circ P \circ S \circ X[K_{10}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
ad347608443ab9c9bbb64f633a5749ab85c45d4174bfd78f6bc79fc4f4ce9ad1
dd71cb2195b1cfab8dcaaf6f3a65c8bb0079847a0800e4427d3a0a815f40a644.
Итерация 11. $K_{11} =$
88ce996c63618e6404a5c8e03ee433854e2ae3eee68991bbbff3c29d38dadb6e
d6a1dae9a6dc6ddf52ce34af272f96d3159c8c624c3fe6e13d695c0bfc89add5,
$L \circ P \circ S \circ X[K_{11}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
a065c55e2168c31576a756c7ecc1a9129cd3d207f8f43073076c30e111fd5f11
9095ca396e9fb78a2bf4781c44e845e447b8fc75b788284aae27582212ec23ee.
Итерация 12. $K_{12} =$
3e0a281ea9bd46063eec550100576f3a506aa168cf82915776b978fccaa32f38
b55f30c79982ca45628e8365d8798477e75a49c68199112a1d7b5a0f7655f2db,
$L \circ P \circ S \circ X[K_{12}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
2a6549f7a5cd2eb4a271a7c71762c8683e7a3a906985d60f8fc86f64e35908b2
9f83b1fe3c704f3c116bdfe660704f3b9c8a1d0531baaffaa3940ae9090a33ab.
Итерация 13. $K_{13} =$
f0b273409eb31aebe432fbae1867212262c848422b6a92f93f6cbab54ed18b83
14b21cffc51e3fa319ff433e76ef6adb0ef9f5e03c907fa1fcf9eca06500bf03,
$X[K_{13}] \circ \ldots \circ L \circ P \circ S \circ X[K_1](m) =$
dad73ab73b7e345f46435c690f05e94a5cb272d242ef44f6b0a4d5d1ad888331
8b31ad01f96e709f08949cd8169f25e09273e8e50d2ad05b5f6de6496c0a8ca8.
Результат выполнения преобразования $g_N(h, m)$**:**
$h =$
203cc15dd55fcaa5b7a3bd98fb2408a67d5b9f33a80bb50540852b204265a2c1
aaca5efe1d8d51b2e1636e34f5becc077d930114fefaf176b69c15ad8f2b6878.
Изменяются значения переменных N **и** Σ**:**
$N =$
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000200,
$\Sigma =$
fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0eeec20f1
20faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ce8f0f2e5e220e5d1.
Длина оставшейся части сообщения меньше 512, поэтому происходит заполнение неполного блока:
$m =$
0000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000001fbe2e5f0eee3c820.
Результат выполнения преобразования $g_N(h, m)$**:**

$h =$
a69049e7bd076ab775bc2873af26f098c538b17e39a5c027d532f0a2b3b56426
c96b285fa297b9d39ae6afd8b9001d97bb718a65fcc53c41b4ebf4991a617227.
Изменяются значения переменных N и Σ:
$N =$
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000240,
$\Sigma =$
fbeafaebef20fffbf0e1e0f0f520e0ed20e8ece0ebe5f0f2f120fff0eeec20f1
20faf2fee5e2202ce8f6f3ede220e8e6eee1e8f0f2d1202ee4d3d8d6d104adf1.
Результат выполнения преобразования $g_0(h, N)$:
$h =$
aee3bd55ea6f387bcf28c6dcbdbbfb3ddacc67dcc13dbd8d548c6bf808111d4b
75b8e74d2afae960835ae6a5f03575559c9fd839783ffcd5cf99bd61566b4818.
Результат выполнения преобразования $g_0(h, \Sigma)$:
$h =$
508f7e553c06501d749a66fc28c6cac0b005746d97537fa85d9e40904efed29d
c345e53d7f84875d5068e4eb743f0793d673f09741f9578471fb2598cb35c230.
Хеш-кодом сообщения M_2 является значение:
$H(M_2) =$
508f7e553c06501d749a66fc28c6cac0b005746d97537fa85d9e40904efed29d.