

Изменение № 1 O'z DSt 1106:2009 Информационная технология.
Криптографическая защита информации. Функция хэширования

Утверждено и введено в действие постановлением Узбекского агентства стандартизации, метрологии и сертификации (агентство «Узстандарт») от 04.07.2014 № 05556

Дата введения 14.07.2014

Срок действия государственного стандарта продлить до 15.10.2019 г.

Раздел 2

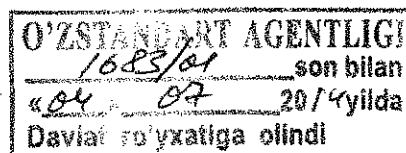
В обозначении государственного стандарта O'z DSt 1109 год утверждения «2006» заменить на «2013».

Раздел 4

В последнем абзаце исключить библиографическую ссылку [1].

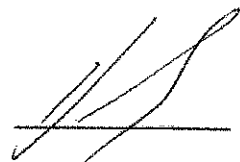
Библиография

Исключить.

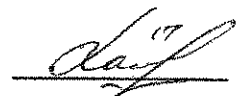


Изменение № 1 O'z DSt 1106:2009

Вр.и.о. директора
ГУП «UNICON.UZ»

 **Х. Хасанов**

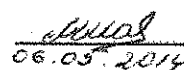
Начальник
научно-исследовательского отдела
криптографии

 **О. Ахмедова**

Младший научный сотрудник
научно-исследовательского отдела
криптографии

 **Д. Джаматова**

Нормоконтроль

 **Л. Шаймарданова**
06.05.2014

СОГЛАСОВАНО

Начальник отдела
информационной безопасности
Государственного комитета связи,
информатизации и информационных
технологий Республики Узбекистан

А.Гафуров
письмо от 25.03.2014 г.
№ 14-8/1679

СОГЛАСОВАНО

Службой национальной
безопасности Республики Узбекистан

письмо от 24.06.2014 г.
№ 39/2199

STANDARTLASHTIRISH, DAVLAT
NAZORATINI MUVOFIQLASHTIRISH VA
AXBOROT TEXNOLOGIYALARINI
JORIY ETISH BOLSHQARMASI

ГОСУДАРСТВЕННЫЙ СТАНДАРТ УЗБЕКИСТАНА

**Информационная технология
КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ
Функция хэширования**

Издание официальное

Узбекское агентство стандартизации, метрологии и сертификации
Ташкент

Предисловие

1 РАЗРАБОТАН И ВНЕСЕН Государственным унитарным предприятием Центр научно-технических и маркетинговых исследований - «UNICON.UZ» (ГУП «UNICON.UZ») Узбекского агентства связи и информатизации

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ постановлением Узбекского агентства стандартизации, метрологии и сертификации (агентство «Узстандарт») от 28.09.09 № 05-163

3 В настоящем стандарте реализованы нормы законов Республики Узбекистан «Об электронной цифровой подписи» и «Об электронном документообороте»

4 ВЗАМЕН O'z DSt 1106:2006

Информация о введении в действие (прекращении действия) настоящего стандарта и изменений к нему на территории Узбекистана публикуется в указателе, издаваемом агентством «Узстандарт». В случае пересмотра или отмены настоящего стандарта соответствующая информация будет опубликована в информационном указателе, издаваемом агентством «Узстандарт».

Исключительное право официального опубликования настоящего стандарта на территории Узбекистана принадлежит агентству «Узстандарт»

Содержание

1	Область применения.	1
2	Нормативные ссылки.	1
3	Термины, определения и обозначения.	2
3.1	Термины и определения.	2
3.2	Обозначения.	2
4	Общие положения.	3
5	Математические соглашения по алгоритму 1.	4
5.1	Математические определения и предпосылки.	4
5.2	Формы представления и соглашения.	5
5.3	Параметры и преобразования функции хэширования данных.	9
6	Алгоритм 1. Основные процессы.	10
6.1	Псевдокод процедуры хэширования.	10
6.2	Преобразования.	12
7	Алгоритм 2. Основные процессы.	16
7.1	Шаговая функция хэширования.	16
7.2	Процедура вычисления хэш-функции.	17
	Приложение А (справочное) Контрольный пример по алгоритму 1.	19
	Приложение В (обязательное) Линейный массив сжатия порядка 1x256.	33
	Приложение С (справочное) Контрольный пример по алгоритму 2.	34
	Библиография.	39

O'ZSTANDART AGENTLIGI
 STANDARTLASHTIRISH,
 MUVOFIQLASHTIRISH
 VA DAVLAT NAZORATI
 DOPHOARMASI

ГОСУДАРСТВЕННЫЙ СТАНДАРТ УЗБЕКИСТАНА**Ахборот технологияси
АХБОРОТНИНГ КРИПТОГРАФИК МУҲОФАЗАСИ
Хэшлаш функцияси****Информационная технология
КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ
Функция хэширования****Information technology
CRYPTOGRAPHIC DATA SECURITY
Hashing function**Дата введения 15.10.2009
15.10.2014**1 Область применения**

Настоящий стандарт определяет алгоритм и процедуру вычисления функции хэширования (далее ФХ) для любой последовательности двоичных символов, которые применяются в криптографических методах обработки и защиты информации, в том числе для реализации процедур электронной цифровой подписи (далее ЭЦП) при передаче, обработке и хранении информации в автоматизированных системах.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритмы криптографического преобразования

O'z DSt 1047:2003 Информационные технологии. Термины и определения

O'z DSt 1109:2006 Информационная технология. Криптографическая защита информации. Термины и определения

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочного стандарта на территории Узбекистана по соответствующему указателю стандартов, составленному по состоянию на 1 января текущего года, и по соответствующим информационным указателям, опубликованным в текущем году. Если ссылочный документ заменен (изменен), то при пользовании настоящим стандартом следует руководствоваться замененным (измененным) стандартом. Если ссылочный документ отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины, определения и обозначения

3.1 Термины и определения

В настоящем стандарте применяются термины по O'z DSt 1047, O'z DSt 1109, а также следующие термины с соответствующими определениями:

3.1.1 модуль: - целое число, используемое для вычисления остатка от деления целого числа на другое целое число в модульной арифметике.

3.1.2 *holat*: Промежуточный результат хэширования.

3.2 Обозначения

В настоящем стандарте использованы следующие обозначения:

- в алгоритме 1:

M – исходные данные (сообщение);

h – хэш-функция;

m – значение хэш-функции, где $m = h(M)$;

k – ключ хэширования;

k_e – этапный ключ в виде двумерного массива порядка 4×8 ;

holat – промежуточный результат хэширования в виде двумерного массива порядка 4×8 ;

holatn – входной блок в виде двумерного массива порядка 4×8 ;

p – модуль, где $p \in \{16, 256\}$;

e – число этапов процедуры хэширования;

b – число блоков в исходных данных;

$\oplus$ – символ операции *XOR* (операции сложения по модулю 2);

$\otimes$ – символ операции умножения диаматриц по модулю p ;

$\circledast$ – символ операции умножения с параметром по модулю p ;

-1 – символ операции обращения по модулю p ;

$\cdot^{-1}$ – символ операции обращения с параметром по модулю p ;

lx – символ операции возведения в степень x с параметром по модулю p ;

ΦX – функция хэширования;

NY – контрольная сумма;

$||$ – символ конкатенации;

- в алгоритме 2:

B^* – множество всех конечных слов в алфавите $B=\{0,1\}$. Чтение слов и нумерация знаков алфавита (символов) осуществляются справа налево (номер правого символа в слове равен единице, второго справа – двум и т.д.);

$|A|$ – длина слова $A \in B^*$;

$V_k(2)$ – множество всех бинарных слов длины k ;

$A||B$ – конкатенация слов $A, B \in B^*$ – слово длины $|A|+|B|$, в котором левые $|A|$ символов образуют слово A , а правые $|B|$ символов образуют слово B . Можно также использовать обозначение $A||B = AB$;

A^k – конкатенация k экземпляров слова A ($A \in B^*$);

$\langle N \rangle_k$ – слово длины k , содержащее двоичную запись вычета $N \pmod{2^k}$ неотрицательного целого числа N ;

$\hat{A}$ – неотрицательное целое число, имеющее двоичную запись A ($A \in B^*$);

$\oplus$ – символ операции *XOR* (операции сложения по модулю 2);

$\oplus'$ – сложение по правилу $A \oplus' B = \langle \hat{A} + \hat{B} \rangle_k$, ($k = |A| = |B|$);

M – исходные данные (сообщение);

h – хэш-функция, отображающая последовательность $M \in B^*$ в слово $h(M) \in V_{256}(2)$;

$h(M)$ – хэш-значение;

$E_k(A)$ – результат зашифрования слова A на ключе K с использованием алгоритма шифрования по ГОСТ 28147 в режиме простой замены ($K \in V_{256}(2)$, $A \in V_{64}(2)$);

H – стартовый вектор хэширования;

$e := g$ – присвоение параметру e значения g .

4 Общие положения

Функция хэширования представляет собой отображение, на вход которого подается сообщение произвольной длины M , а выходом является строка фиксированной длины $h(M)$.

Хэш-значение $h(M)$ – это сжатое двоичное представление сообщения M произвольной длины. Функция хэширования позволяет сжать подписываемое сообщение M до нескольких десятков или сотен бит.

Базовые требования к криптографической хэш-функции:

- на вход функции может подаваться сообщение любой длины;
- на выходе функции получается сообщение фиксированной длины;
- хэш-функция достаточно просто вычисляется для любого сообщения;

- хэш-функция – однонаправленная функция;

- зная сообщение M трудно определить, другое сообщение M' , для которого $h(M) = h(M')$.

Принятая за основу функция хэширования [1], определяется на основе алгоритма 2 вычисления шаговой функции хэширования x (т.е. отображения $x : V_{256}(2) \times V_{256}(2) \rightarrow V_{256}(2)$) итеративной процедуры вычисления значения хэш-функции h .

O'Z STANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI

5 Математические соглашения по алгоритму 1

5.1 Математические определения и предпосылки

5.1.1 Для определения ФХ необходимо описать базовые математические объекты, используемые в процессе хэширования. В данном разделе установлены основные математические определения и требования, накладываемые на параметры алгоритма хэширования данных.

5.1.2 Первоначальный этапный ключ k_e формируется из линейного массива ключа хэширования k на полубайтовом (байтовом) уровне, второй полубайт (байт) которого должен быть четным натуральным числом, выбираемым из диапазона от 4 до 10 (10 до 120). Каждая часть квадратной формы массива первоначального этапного ключа k_e должна отвечать требованию обратимости. Для этого определитель результата диапреобразования каждой из частей квадратной формы массива k_e должен быть отличным от нуля.

5.1.3 В функции хэширования используется односторонняя функция модульной арифметики, вычисления по которой осуществляются легко на том же уровне трудоемкости, что и в операциях возведения в степень, а инвертирование (обращение) функции требует не меньших вычислительных затрат и времени, чем в процессе решения проблемы дискретного логарифма при неизвестном параметре (A, B) . Главные операции умножения, возведения в степень и обращения в новой односторонней функции названы умножением, возведением в степень и обращением с параметром. Односторонняя функция возведения в степень является частным случаем данной односторонней функции. В функции хэшировании в качестве параметра (коэффициента) используется тройка натуральных чисел (A, B, R) .

5.1.4 Операция возведения основания a в степень x с параметром по модулю p обозначается в виде:

$$a^{lx} \pmod{p}. \quad (1)$$

Например, для $x = -3$ с параметром (A, B, R) имеем:

$$a^{l-3} \equiv ((a^{l-1})^{l-1})^{l-1} \pmod{p}, \quad (2)$$

где:

$$a^{l-2} \equiv (a^{l-1})^{l-1} \pmod{p}, \quad a^{l-3} \equiv (a^{l-2})^{l-1} \pmod{p}. \quad (3)$$

Операция обращения переменной a с параметром (A, B, R) по модулю p , обозначается в виде: a^{l-1} и определяется:

$$a^{l-1} \equiv -Aa(B + Ra)^{-1} \pmod{p}, \quad (4)$$

где: $a \otimes a^{l-1} \equiv 0 \pmod{p}$,

$a^{l-1} \otimes a \neq 0 \pmod{p}$, 0 – единичный элемент алгебры параметров.

Свойство $a^{l-1} \otimes a \neq 0 \pmod{p}$ служит для обеспечения однонаправленности функции возведения основания a в отрицательную степень $x = -3$ с закрытым параметром (A, B, R) , при различных его значениях для различных степеней.

Операция умножения чисел a и b с параметром (A, B, R) по модулю p , определяется:

$$a \circledast b \equiv Aa + (B + Ra)b \pmod{p}. \quad (5)$$

В функции хэширования используется операция умножения квадратных диаматриц по модулю p , которая приведена в 6.2. Использование операции умножения диаматриц при изменении одного элемента входной диаматрицы позволяет увеличить количество измененных элементов результата операции в 1,5 - 1,75 раза в сравнении с операцией умножения матриц.

Примечание – В приводимых математических выражениях модуль $p \in \{16, 256\}$. При $p=16$ и при длинах ключа хэширования и входного блока 128 bit функция хэширования используется как ключевая функция хэширования. При $p=256$ и при длинах ключа хэширования и входного блока 256 bit функция хэширования используется как бесключевая функция хэширования.

5.2 Формы представления и соглашения

5.2.1 Входы и выходы

5.2.1.1 В ФХ длина входной последовательности кратна 128 или 256 bit, выходная последовательность и ключ хэширования имеют фиксированные длины 128 или 256 bit. Другие длины входа, выхода и ключа хэширования настоящим стандартом не допускаются.

5.2.1.2 Биты в данных последовательностях нумеруются, начиная с нуля и заканчивая номером на единицу меньшим, чем длина последовательности. При этом i - порядковый номер бита, известный как его индекс, будет находиться в диапазоне $0 \leq i < 128$ или $0 \leq i < 256$.

5.2.1.3 В процедуре хэширования данных алгоритма ФХ используются этапные ключи, которые формируются на основе ключа хэширования и промежуточного результата хэширования.

5.2.2 Полубайты и байты

5.2.2.1 Базовой единицей для обработки входных блоков длиной 128 bit в алгоритме ФХ является полубайт - последовательность из четырёх бит. Битовые последовательности входа, выхода, ключа хэширования и этапного ключа обрабатываются как массивы полубайтов, которые формируются путем деления указанных битовых последовательностей на группы из четырёх смежных бит.

5.2.2.2 Для входа, выхода или ключа хэширования, обозначенных буквой a , полубайты в результирующем массиве будут обозначаться с использованием одной из двух форм записи - a_n или $a[n]$, где значение n находится в диапазоне $0 \leq n < 32$.

5.2.2.3 Все значения полубайта представляются как конкатенация его индивидуальных битовых значений (0 или 1) между скобками в порядке $\{b_3, b_2, b_1, b_0\}$.

Например, для значения полубайта $\{0011\}$, битовые значения определяются: $b_3 = 0, b_2 = 0, b_1 = 1, b_0 = 1$.

O'Z STANDART AGENTLIGI
STANDARTLASHTIRISH.
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI 5
BOSHDARMANI

5.2.2.4 Базовой единицей для обработки входных блоков длиной 256 bit в алгоритме ФХ является байт - последовательность из восьми бит. Битовые последовательности входа, выхода, ключа хэширования и этапного ключа обрабатываются как массивы байтов, которые формируются путем деления указанных битовых последовательностей на группы из восьми смежных бит.

5.2.2.5 Все значения байта представляются как конкатенация его индивидуальных битовых значений (0 или 1) между скобками в порядке $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$.

Например, для значения байта $\{01100011\}$, битовые значения определяются как $b_7=0, b_6=1, b_5=1, b_4=0, b_3=0, b_2=0, b_1=1, b_0=1$.

5.2.3 Массивы полубайтов и байтов

5.2.3.1 Массивы полубайтов для входных блоков длиной 128 bit представляются в следующей форме:

$a_0, a_1, a_2, a_3, \dots, a_{31}$.

Полубайты и расположение битов в пределах полубайтов определяются из 128 - битовой входной последовательности

$kir_0, kir_1, kir_2, kir_3, \dots, kir_{126}, kir_{127}$

следующим образом:

$a_0 = \{kir_0, kir_1, kir_2, kir_3\};$

$a_1 = \{kir_4, kir_5, kir_6, kir_7\};$

:

$a_{31} = \{kir_{124}, kir_{125}, kir_{126}, kir_{127}\}.$

С учетом выше изложенного на рисунке 1 показано, как нумеруются биты в каждом полубайте.

Последовательность битов	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Номер полубайта	0				1				2				3			
Номер бита в полубайте	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0

Рисунок 1- Номера полубайтов и битов

5.2.3.2 Массивы байтов для входных блоков длиной 256 bit представляются в следующей форме:

$a_0, a_1, a_2, a_3, \dots, a_{31}$.

Байты и расположение битов в пределах байтов определяются из 256 - битовой входной последовательности

$kir_0, kir_1, kir_2, kir_3, \dots, kir_{254}, kir_{255}$

следующим образом:

$a_0 = \{kir_0, kir_1, \dots, kir_7\};$

$$a_1 = \{kir_8, kir_9, \dots, kir_{15}\};$$

:

$$a_{31} = \{kir_{247}, kir_{248}, \dots, kir_{255}\}.$$

С учетом вышеизложенного на рисунке 2 показано, как нумеруются биты в каждом байте.

Последовательность битов	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Номер байта	1								1								2							
Номер бита в байте	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Рисунок 2- Номера байтов и битов

5.2.4 Массив *holat*

5.2.4.1 Процесс обработки данных в алгоритме 1 ФХ можно интерпретировать как выполнение операций алгоритма над двумерным массивом полубайтов (байтов) *holat*.

Массив *holat* состоит из четырех строк (рядов) и восьми столбцов полубайтов (байтов), причем каждая строка содержит 32 (64) bit.

В массиве *holat*, обозначаемом *h*, каждый отдельный полубайт (байт) имеет два индекса *s* и *u*, где *s* - номер его строки в диапазоне $0 \leq s < 4$; *u* - номер его столбца в диапазоне $0 \leq u < 8$. Эта индексация позволяет ссылаться на конкретный полубайт (байт) массива *holat* как на *h[s, u]*.

5.2.4.2 В самом начале процесса хэширования, очередной блок входного массива *kir* - массив полубайтов (байтов) *kir₀, kir₁, ... kir₃₁* - копируется в массив *holatn* и формируется массив *holat*, результат применения операции сложения по модулю 2 к массиву *holatn* и результату хэширования предыдущего блока. Затем над массивом *holat* выполняются необходимые операции хэширования, после чего окончательное значение элементов массива *holat* копируется в выходной массив *chiq* - массив полубайтов (байтов) *chiq₀, chiq₁, ... chiq₃₁*, в соответствии с рисунком 3.

Примечание - Стандарт допускает в процессе преобразования линейных массивов *kir*, *chiq* расположить элементы по столбцам (рисунок 3) или строкам массива 4x8 порядка.

Следует заметить, что перед преобразованиями, основанными на использовании операции умножения матриц, двумерный массив *holat[4,8]* следует разбить на левую и правую половины (части): *H_{ch}[4,4]* и *H_o[4,4]*, в соответствии с рисунком 4.

По окончании преобразования эти половины вновь копируются в массив *holat[4,8]*.

U'ZSTANDART ADOPTETIL
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI

5.2.2.4 Базовой единицей для обработки входных блоков длиной 256 bit в алгоритме ФХ является байт - последовательность из восьми бит. Битовые последовательности входа, выхода, ключа хэширования и этапного ключа обрабатываются как массивы байтов, которые формируются путем деления указанных битовых последовательностей на группы из восьми смежных бит.

5.2.2.5 Все значения байта представляются как конкатенация его индивидуальных битовых значений (0 или 1) между скобками в порядке $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$.

Например, для значения байта $\{01100011\}$, битовые значения определяются как $b_7=0, b_6=1, b_5=1, b_4=0, b_3=0, b_2=0, b_1=1, b_0=1$.

5.2.3 Массивы полубайтов и байтов

5.2.3.1 Массивы полубайтов для входных блоков длиной 128 bit представляются в следующей форме:

$a_0, a_1, a_2, a_3, \dots, a_{31}$.

Полубайты и расположение битов в пределах полубайтов определяются из 128 - битовой входной последовательности

$kir_0, kir_1, kir_2, kir_3, \dots, kir_{126}, kir_{127}$

следующим образом:

$a_0 = \{kir_0, kir_1, kir_2, kir_3\};$

$a_1 = \{kir_4, kir_5, kir_6, kir_7\};$

:

$a_{31} = \{kir_{124}, kir_{125}, kir_{126}, kir_{127}\}.$

С учетом выше изложенного на рисунке 1 показано, как нумеруются биты в каждом полубайте.

Последовательность битов	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Номер полубайта	0				1				2				3			
Номер бита в полубайте	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0

Рисунок 1- Номера полубайтов и битов

5.2.3.2 Массивы байтов для входных блоков длиной 256 bit представляются в следующей форме:

$a_0, a_1, a_2, a_3, \dots, a_{31}$.

Байты и расположение битов в пределах байтов определяются из 256 - битовой входной последовательности

$kir_0, kir_1, kir_2, kir_3, \dots, kir_{254}, kir_{255}$

следующим образом:

$a_0 = \{kir_0, kir_1, \dots, kir_7\};$

$$a_1 = \{kir_8, kir_9, \dots, kir_{15}\};$$

:

$$a_{31} = \{kir_{247}, kir_{248}, \dots, kir_{255}\}.$$

С учетом вышеизложенного на рисунке 2 показано, как нумеруются биты в каждом байте.

Последовательность битов	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	23
Номер байта	1								1								2				
Номер бита в байте	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	0

Рисунок 2- Номера байтов и битов

5.2.4 Массив *holat*

5.2.4.1 Процесс обработки данных в алгоритме 1 ФХ можно интерпретировать как выполнение операций алгоритма над двумерным массивом полубайтов (байтов) *holat*.

Массив *holat* состоит из четырех строк (рядов) и восьми столбцов полубайтов (байтов), причем каждая строка содержит 32 (64) bit.

В массиве *holat*, обозначаемом *h*, каждый отдельный полубайт (байт) имеет два индекса *s* и *u*, где *s* - номер его строки в диапазоне $0 \leq s < 4$; *u* - номер его столбца в диапазоне $0 \leq u < 8$. Эта индексация позволяет ссылаться на конкретный полубайт (байт) массива *holat* как на *h[s, u]*.

5.2.4.2 В самом начале процесса хэширования, очередной блок входного массива *kir* - массив полубайтов (байтов) *kir₀, kir₁, ..., kir₃₁* копируется в массив *holatn* и формируется массив *holat*, результат применения операции сложения по модулю 2 к массиву *holatn* и результату хэширования предыдущего блока. Затем над массивом *holat* выполняются необходимые операции хэширования, после чего окончательное значение элементов массива *holat* копируется в выходной массив *chiq* - массив полубайтов (байтов) *chiq₀, chiq₁, ..., chiq₃₁*, в соответствии с рисунком 3.

Примечание – Стандарт допускает в процессе преобразования линейных массивов *kir*, *chiq* расположить элементы по столбцам (рисунок 3) или строкам массива 4x8 порядка.

Следует заметить, что перед преобразованиями, основанными на использовании операции умножения матриц, двумерный массив *holat[4,8]* следует разбить на левую и правую половины (части): *H_{ch}[4,4]* и *H_c[4,4]*, в соответствии с рисунком 4.

По окончании преобразования эти половины вновь копируются в массив *holat[4,8]*.

Полубайты (байты) массива *kir*

<i>kir</i> ₀	<i>kir</i> ₄	<i>kir</i> ₈	<i>kir</i> ₁₂	<i>kir</i> ₁₆	<i>kir</i> ₂₀	<i>kir</i> ₂₄	<i>kir</i> ₂₈
<i>kir</i> ₁	<i>kir</i> ₅	<i>kir</i> ₉	<i>kir</i> ₁₃	<i>kir</i> ₁₇	<i>kir</i> ₂₁	<i>kir</i> ₂₅	<i>kir</i> ₂₉
<i>kir</i> ₂	<i>kir</i> ₆	<i>kir</i> ₁₀	<i>kir</i> ₁₄	<i>kir</i> ₁₈	<i>kir</i> ₂₂	<i>kir</i> ₂₆	<i>kir</i> ₃₀
<i>kir</i> ₃	<i>kir</i> ₇	<i>kir</i> ₁₁	<i>kir</i> ₁₅	<i>kir</i> ₁₉	<i>kir</i> ₂₃	<i>kir</i> ₂₇	<i>kir</i> ₃₁

↓ Массив *hn*

<i>hn</i> [0,0]	<i>hn</i> [0,1]	<i>hn</i> [0,2]	<i>hn</i> [0,3]	<i>hn</i> [0,4]	<i>hn</i> [0,5]	<i>hn</i> [0,6]	<i>hn</i> [0,7]
<i>hn</i> [1,0]	<i>hn</i> [1,1]	<i>hn</i> [1,2]	<i>hn</i> [1,3]	<i>hn</i> [1,4]	<i>hn</i> [1,5]	<i>hn</i> [1,6]	<i>hn</i> [1,7]
<i>hn</i> [2,0]	<i>hn</i> [2,1]	<i>hn</i> [2,2]	<i>hn</i> [2,3]	<i>hn</i> [2,4]	<i>hn</i> [2,5]	<i>hn</i> [2,6]	<i>hn</i> [2,7]
<i>hn</i> [3,0]	<i>hn</i> [3,1]	<i>hn</i> [3,2]	<i>hn</i> [3,3]	<i>hn</i> [3,4]	<i>hn</i> [3,5]	<i>hn</i> [3,6]	<i>hn</i> [3,7]

↓ Полубайты (байты) массива *h*

<i>h</i> [0,0]	<i>h</i> [0,1]	<i>h</i> [0,2]	<i>h</i> [0,3]	<i>h</i> [0,4]	<i>h</i> [0,5]	<i>h</i> [0,6]	<i>h</i> [0,7]
<i>h</i> [1,0]	<i>h</i> [1,1]	<i>h</i> [1,2]	<i>h</i> [1,3]	<i>h</i> [1,4]	<i>h</i> [1,5]	<i>h</i> [1,6]	<i>h</i> [1,7]
<i>h</i> [2,0]	<i>h</i> [2,1]	<i>h</i> [2,2]	<i>h</i> [2,3]	<i>h</i> [2,4]	<i>h</i> [2,5]	<i>h</i> [2,6]	<i>h</i> [2,7]
<i>h</i> [3,0]	<i>h</i> [3,1]	<i>h</i> [3,2]	<i>h</i> [3,3]	<i>h</i> [3,4]	<i>h</i> [3,5]	<i>h</i> [3,6]	<i>h</i> [3,7]

↓ Полубайты (байты) массива *chiq*

<i>chiq</i> ₀	<i>chiq</i> ₄	<i>chiq</i> ₈	<i>chiq</i> ₁₂	<i>chiq</i> ₁₆	<i>chiq</i> ₂₀	<i>chiq</i> ₂₄	<i>chiq</i> ₂₈
<i>chiq</i> ₁	<i>chiq</i> ₅	<i>chiq</i> ₉	<i>chiq</i> ₁₃	<i>chiq</i> ₁₇	<i>chiq</i> ₂₁	<i>chiq</i> ₂₅	<i>chiq</i> ₂₉
<i>chiq</i> ₂	<i>chiq</i> ₆	<i>chiq</i> ₁₀	<i>chiq</i> ₁₄	<i>chiq</i> ₁₈	<i>chiq</i> ₂₂	<i>chiq</i> ₂₆	<i>chiq</i> ₃₀
<i>chiq</i> ₃	<i>chiq</i> ₇	<i>chiq</i> ₁₁	<i>chiq</i> ₁₅	<i>chiq</i> ₁₉	<i>chiq</i> ₂₃	<i>chiq</i> ₂₇	<i>chiq</i> ₃₁

Рисунок 3 - Вход и выход массива *h*Массив *H_{ch}*

<i>h_{ch}</i> [0,0]	<i>h_{ch}</i> [0,1]	<i>h_{ch}</i> [0,2]	<i>h_{ch}</i> [0,3]
<i>h_{ch}</i> [1,0]	<i>h_{ch}</i> [1,1]	<i>h_{ch}</i> [1,2]	<i>h_{ch}</i> [1,3]
<i>h_{ch}</i> [2,0]	<i>h_{ch}</i> [2,1]	<i>h_{ch}</i> [2,2]	<i>h_{ch}</i> [2,3]
<i>h_{ch}</i> [3,0]	<i>h_{ch}</i> [3,1]	<i>h_{ch}</i> [3,2]	<i>h_{ch}</i> [3,3]

Массив *H_o*

<i>h_o</i> [0,0]	<i>h_o</i> [0,1]	<i>h_o</i> [0,2]	<i>h_o</i> [0,3]
<i>h_o</i> [1,0]	<i>h_o</i> [1,1]	<i>h_o</i> [1,2]	<i>h_o</i> [1,3]
<i>h_o</i> [2,0]	<i>h_o</i> [2,1]	<i>h_o</i> [2,2]	<i>h_o</i> [2,3]
<i>h_o</i> [3,0]	<i>h_o</i> [3,1]	<i>h_o</i> [3,2]	<i>h_o</i> [3,3]

Рисунок 4 - Массивы *H_{ch}*[4,4] и *H_o*[4,4]

5.2.5 Массив этапного ключа *k_e*

5.2.5.1 В начале процедуры хэширования ключ хэширования *k* копируется в массив этапного ключа *k_e*. Каждый этап содержит преобразования, в результате которых претерпевают изменения массивы этапного ключа.

Массив этапного ключа k_e состоит из четырёх строк и восьми столбцов полубайтов (байтов), и на каждый отдельный полубайт (байт) массива можно ссылаться $k_e[s, u]$, аналогично, в массиве *holat* (рисунок 5).

Массив k_e

$k_e[0,0]$	$k_e[0,1]$	$k_e[0,2]$	$k_e[0,3]$	$k_e[0,4]$	$k_e[0,5]$	$k_e[0,6]$	$k_e[0,7]$
$k_e[1,0]$	$k_e[1,1]$	$k_e[1,2]$	$k_e[1,3]$	$k_e[1,4]$	$k_e[1,5]$	$k_e[1,6]$	$k_e[1,7]$
$k_e[2,0]$	$k_e[2,1]$	$k_e[2,2]$	$k_e[2,3]$	$k_e[2,4]$	$k_e[2,5]$	$k_e[2,6]$	$k_e[2,7]$
$k_e[3,0]$	$k_e[3,1]$	$k_e[3,2]$	$k_e[3,3]$	$k_e[3,4]$	$k_e[3,5]$	$k_e[3,6]$	$k_e[3,7]$

Рисунок 5 - Массив этапного ключа k_e

5.2.5.2 Следует заметить, что перед преобразованиями, основанными на использовании операции умножения матриц, двумерный массив $k_e[4,8]$ следует разбить на левую и правую половины (части): $k_{ech}[4,4]$ и $k_{eo}[4,4]$ (рисунок 6).

Массив k_{ech}

$k_{ech}[0,0]$	$k_{ech}[0,1]$	$k_{ech}[0,2]$	$k_{ech}[0,3]$
$k_{ech}[1,0]$	$k_{ech}[1,1]$	$k_{ech}[1,2]$	$k_{ech}[1,3]$
$k_{ech}[2,0]$	$k_{ech}[2,1]$	$k_{ech}[2,2]$	$k_{ech}[2,3]$
$k_{ech}[3,0]$	$k_{ech}[3,1]$	$k_{ech}[3,2]$	$k_{ech}[3,3]$

Массив k_{eo}

$k_{eo}[0,0]$	$k_{eo}[0,1]$	$k_{eo}[0,2]$	$k_{eo}[0,3]$
$k_{eo}[1,0]$	$k_{eo}[1,1]$	$k_{eo}[1,2]$	$k_{eo}[1,3]$
$k_{eo}[2,0]$	$k_{eo}[2,1]$	$k_{eo}[2,2]$	$k_{eo}[2,3]$
$k_{eo}[3,0]$	$k_{eo}[3,1]$	$k_{eo}[3,2]$	$k_{eo}[3,3]$

Рисунок 6 - Массивы $k_{ech}[4,4]$ и $k_{eo}[4,4]$

По окончании преобразования эти половины вновь копируются в массив $k_e[4,8]$.

5.3 Параметры и преобразования функции хэширования данных

ФХ использует следующие параметры и функции:

- k – ключ хэширования с длиной в 128 или 256 bit в виде линейного массива на полубайтовом (байтовом) уровне;
- k_e – этапный (раундовый) ключ в виде двумерного массива порядка 4×8 ;
- b – число блоков в исходных данных;
- uzunlik* – предпоследний блок входных данных в функцию хэширования, содержащий длину исходных данных в битах;
- NY* – последний блок входных данных в функцию хэширования, содержащий сумму значений исходных данных в десятичной системе счисления;
- p – модуль, $p \in \{16, 256\}$;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI

g) e_0 – общее число этапов процедуры хэширования равное $(b+2)10$, для входных 128 и 256 битовых блоков;

h) $Qo'sh(holat, holatn)$ - преобразование, используемое в процедуре хэширования, над элементами на полубайтовом (байтовом) уровне текущего значения массива *holat* и массива *holatn* по модулю p на основе операции возведения в степень с параметром (A, B, R) ;

i) $BaytZichlash(holat, holatn)$ - преобразование, используемое в процедуре хэширования, над элементами на полубайтовом (байтовом) уровне текущего значения массива *holat* и массива *holatn* с использованием операции *XOR*, если модуль $p=16$, или на основе линейного массива сжатия в один массив, если модуль $p=256$;

j) $Aralash(holat, k_e)$ - преобразование, используемое в процедуре хэширования на основе операции умножения диаматриц. Тут преумножаемые диаматрицы взаимно-однозначно соответствуют правым и левым половинам квадратной формы двумерных массивов *holat* и этапного ключа k_e , соответственно;

k) $SurHolat(holat)$ - преобразование, используемое в процедуре хэширования, которое осуществляется над массивом *holat*, заключается в циклических сдвигах всех четырех строк массива *holat* по горизонтали и по вертикали на различные значения сдвигов;

l) $SurKalit(k_e)$ - преобразование, используемое в процедуре хэширования, которое осуществляется над массивом k_e , заключается в циклических сдвигах всех четырех строк массива k_e по горизонтали и по вертикали на различные значения сдвигов;

m) $TuzilmaKalit(k_e, k)$ - преобразование, используемое в конце каждого этапа процедуры хэширования, которое осуществляется над каждым полубайтом (байтом) массива k_e с целью приведения её структуры к структуре исходного ключа хэширования k ; результат данного преобразования удовлетворяет условиям обратимости каждой из квадратных частей массива k_e .

6 Алгоритм 1. Основные процессы

6.1 Псевдокод процедуры хэширования

6.1.1 Хэшируемое сообщение разбивается на b блоков длиной 128 (256) bit, т.е. $1, \dots, b$ блоки на полубайтовом (байтовом) уровне, а недостающая часть последнего (b -го) блока заполняется последовательностью полубайтов (байтов) из нулей - "0"; в результате определяется основная часть входных данных функции хэширования состоящая из b блоков; вычисляется общая длина основной части в бит по модулю 2^{256} , который представляет собой блок *uzunlik* длиной в 256 bit; далее вычисляется сумма значений блоков основной части по модулю 2^{256} , которая представляет собой блок *контрольной суммы (NY)* длиной в 256 bit; основная часть, блок *uzunlik* и блок *NY* в форме двумерных

массивов элементов на полубайтовом (байтовом) уровне из $b+2$ блоков представляют собой входные данные функции хэширования.

Начальный этап завершается копированием ключа хэширования k длиной 128 (256) bit в двумерный массив k_e .

Процессы хэширования в отношении к каждому блоку входных данных начинаются с выполнения цепочки пары преобразований *Qo'sh (holat, holatn)*, *BaytZichlash (holat, holatn)* над двумя блоками *holat* и *holatn* и завершается формированием в течении 10 этапов текущего хэш-значения *holat*. В самом начале процессов хэширования в роли начального хэш-значения используется 1 блок как блок *holat*, а 2 блок - как блок *holatn*; в случае, если входные данные состоят только из одного блока, то в качестве 2 блока используется блок *uzunlik*.

Затем в массив *holatn* копируется очередной блок и над результатом пары преобразований *Qo'sh (holat, holatn)*, *BaytZichlash (holat, holatn)*, над текущим хэш-значением *holat* и *holatn* реализуется 10 этапов процедуры хэширования и т.д. Последним блоком, копируемым в массив *holatn*, является блок *NY*. Таким образом, общее количество этапов ФХ равняется $(b+1)10$.

6.1.2 Каждый этап (раунд) процедуры хэширования относительно к блокам, наряду с начальной парой преобразований *Qo'sh (holat, holatn)*, *BaytZichlash (holat, holatn)*, содержит преобразования *Aralash(holat, k_e)*, *Qo'sh(holat, holatn)*, *SurHolat(holat)*, *SurKalit(k_e)*, *TuzilmaKalit(k_e, k)*, осуществляемые в циклическом порядке.

6.1.3 Псевдокод процедуры хэширования для входных блоков длиной 256 bit приведен на рисунке 7. Для входных блоков длиной 128 bit псевдокод процедуры хэширования отличается лишь способом задания массивов на полубайтовом уровне. Недостающая часть последнего (b -го) блока основной части при вычислении блока *uzunlik* не учитывается. Отдельные преобразования *Qo'sh (holat, holatn)*, *BaytZichlash (holat, holatn)*, *Aralash(holat, k_e)*, *SurHolat(holat)*, *SurKalit(k_e)*, *TuzilmaKalit(k_e, k)*, используемые в процедуре хэширования, приведены в 6.2.

O'ZSTATISTIK AGENTLIGI
STATISTIKALASHTIRISH,
MONITORING
VA DAVLAT HAZIRATI
BOSQANMAQI

```

Hesh (byte kirish [32*(b+2)], byte chiqish [32], byte k[32], byte zichlash [256], byte b)
Begin
    byte ke [4,8], holat [4,8], holatn[4,8], uzunlik[4,8], NY[4,8]
    R=k[1], holat = kirish (1-blok), ke= k
    For i =1 step 1 to ( b +1)
        holatn = kirish((i+1)-blok)
        Qo'sh(holat, holatn)
        BaytZichlash (holat, holatn)
        For j = 1 step 1 to 10
            Aralash(holat, ke)
            holatn= ke
            Qo'sh(holat, holatn)
            ke = holatn
            SurHolat(holat)
            SurKalit(ke)
            If ( j =8) then
                holatn= kirish((i+1)-blok)
                Qo'sh(holat, holatn)
                BaytZichlash (holat, holatn)
            Else
                holatn= ke
                Qo'sh(holat, holatn)
                ke = holatn
            End If
            TuzilmaKalit(ke,k)
            If ( j =9) then
                holatn= kirish((i+1)-blok)
                Qo'sh(holat, holatn)
                BaytZichlash (holat, holatn)
            End If
        End For
        holat = holatn
    End For
    chiqish= holat
End

```

Рисунок 7 - Псевдокод процедуры хэширования (по алгоритму 1)

6.2 Преобразования

6.2.1 Преобразование *Qo'sh(holat, holatn, R)*

В преобразовании *Qo'sh(holat, holatn, R)* каждый полубайт (байт) массива *holat* и *holatn* заменяется на его $x = -3$ степень по модулю p на основе операции возведения в степень с параметром (A, B, R) при $R = k[1]$.

Над каждым полубайт (байт)ом массива *holat* и *holatn* выполняются следующие преобразования:

для $0 \leq s < 4, 0 \leq u < 8$,

$$h[s,u]^{-1} \equiv -A_{1h}h[s,u](B_{1h} + Rxh[s,u])^{-1} \pmod{p},$$

$$hn[s,u]^{-1} \equiv -A_{1hn}hn[s,u](B_{1hn} + Rhn[s,u])^{-1} \pmod{p},$$

где:

$$A_{1h} \equiv J_{1hn} + 2hn[s,u] + \Delta \pmod{p},$$

$$A_{1hn} \equiv J_{1h} + 2h[s,u] + \Delta \pmod{p},$$

$$B_{1h} \equiv K_{1h} - 2h[s,u] \pmod{p},$$

$$B_{1hn} \equiv K_{1hn} - 2hn[s,u] \pmod{p},$$

$$J_{1h} \equiv \sum_{s=3,u=7}^{s=0,u=0} h[s,u] \pmod{p},$$

$$K_{1h} \equiv \prod_{s=3,u=7}^{s=0,u=0} h[s,u] + \Delta \pmod{p},$$

$$J_{1hn} \equiv \sum_{s=3,u=7}^{s=0,u=0} hn[s,u] \pmod{p},$$

$$K_{1hn} \equiv \prod_{s=3,u=7}^{s=0,u=0} hn[s,u] + \Delta \pmod{p}.$$

В вышеприведенных выражениях, если:

- для $A_{1h} J_{1hn} \pmod{2} \equiv 1$ или для $A_{1hn} J_{1h} \pmod{2} \equiv 1$, то принимается $\Delta=0$, иначе принимается $\Delta=1$;

- для $K_{1h} h[s,u] \pmod{2} \equiv 0$, или для $K_{1hn} hn[s,u] \pmod{2} \equiv 0$, то принимается $h[s,u] \equiv h[s,u]+1 \pmod{p}$ и $hn[s,u] \equiv hn[s,u]+1 \pmod{p}$ соответственно;

- для $K_{1h} \prod_{s=3,u=7}^{s=0,u=0} h[s,u] \not\equiv \pm (J_{1hn}+1) \pmod{p}$ или для $K_{1hn} \prod_{s=3,u=7}^{s=0,u=0} hn[s,u] \not\equiv \pm (J_{1h}+1) \pmod{p}$, то принимается $\Delta=0$, иначе принимается $\Delta=10$,

где: h – элемент массива *holat*, hn – элемент массива *holatn*.

$$h[s,u]^{-2} \equiv -A_{2h}h[s,u]^{-1}(B_{2h} + Rh[s,u]^{-1})^{-1} \pmod{p},$$

Если $h[s,u]^{-2} \equiv 0 \pmod{p}$, то вычисляется $h[s,u]^{-2} \equiv A_{2h} + B_{2h} \pmod{p}$,

$$hn[s,u]^{-2} \equiv -A_{2hn}hn[s,u]^{-1}(B_{2hn} + Rhn[s,u]^{-1})^{-1} \pmod{p},$$

Если $hn[s,u]^{-2} \equiv 0 \pmod{p}$, то вычисляется $hn[s,u]^{-2} \equiv A_{2hn} + B_{2hn} \pmod{p}$,

$$h[s,u]^{-3} \equiv -A_{2h}h[s,u]^{-2}(B_{2h} + Rh[s,u]^{-2})^{-1} \pmod{p},$$

$$hn[s,u]^{-3} \equiv -A_{2hn}hn[s,u]^{-2}(B_{2hn} + Rhn[s,u]^{-2})^{-1} \pmod{p},$$

где:

$$A_{2h} \equiv J_{1hn} + J_{2hn} + 2hn[s,u]^{-1} + \Delta \pmod{p},$$

Если $A_{2h} \equiv 0 \pmod{p}$, то принимается $A_{2h} = 1$,

$$A_{2hn} \equiv J_{1h} + J_{2h} + 2h[s,u]^{-1} + \Delta \pmod{p},$$

Если $A_{2hn} \equiv 0 \pmod{p}$, то принимается $A_{2hn} = 1$,

$$B_{2h} \equiv K_{1h}K_{2h} - 2h[s,u]^{-1} \pmod{p},$$

$$B_{2hn} \equiv K_{1hn}K_{2hn} - 2hn[s,u]^{-1} \pmod{p},$$

$$J_{2h} \equiv \sum_{s=3,u=7}^{s=0,u=0} h[s,u]^{-1} \pmod{p},$$

$$K_{2h} \equiv \prod_{s=3,u=7}^{s=0,u=0} h[s,u]^{-1} + \Delta \pmod{p}.$$

$$J_{2hn} \equiv \sum_{s=3,u=7}^{s=0,u=0} hn[s,u]^{-1} \pmod{p},$$

$$K_{2hn} \equiv \prod_{s=3,u=7}^{s=0,u=0} hn[s,u]^{-1} + \Delta \pmod{p}.$$

В вышеприведенных выражениях, если:

- для A_{2h} или $A_{2hn} J_{1h}+J_{2h} \pmod{2} \equiv 1$, $J_{1hn}+J_{2hn} \pmod{2} \equiv 1$, соответственно, то принимается $\Delta=0$, иначе принимается $\Delta=1$;

U'Z DSt 1106:2009
STANDETLASHTIRISH,
NUQDIFOLASHTIRISH
YA BAYLAT NAZORATI
DOIRATIDIR

- для $K_{2h} \quad h[s,u]^{l-1} \pmod{2} \equiv 0$, или для $K_{2hn} \quad hn[s,u]^{l-1} \pmod{2} \equiv 0$, то принимается $h[s,u]^{l-1} \equiv h[s,u]^{l-1} + 1 \pmod{p}$ или $hn[s,u]^{l-1} \equiv hn[s,u]^{l-1} + 1 \pmod{p}$ соответственно;

- для $K_{2h} \prod_{s=3,u=7}^{s=3,u=7} h[s,u]^{l-1} \not\equiv \pm (J_{2hn} + 1) \pmod{p}$ или для $K_{2hn} \prod_{s=3,u=7}^{s=3,u=7} hn[s,u]^{l-1} \not\equiv \pm (J_{2h} + 1) \pmod{p}$, то принимается $\Delta = 0$, иначе принимается $\Delta = 10$.

Результаты преобразований копируются в массивы *holat* и *holatn*.

6.2.2 Преобразование *BaytZichlash (holat, holatn)*

Если длина ключа хэширования и длина входного блока составляют 128 bit, тогда в преобразовании *BaytZichlash (holat, holatn)* каждый полубайт массива *holatn* прибавляется к одноименному с ним полубайту массива *holat* с помощью операции простого побитового сложения *XOR* (сложения по модулю 2). Массив *holatn* состоит из восьми полуслов. Эти полуслова суммируются по отдельности с элементами столбцов массива *holat* так, что

$$\begin{aligned} & [h'[0,u], h'[1,u], h'[2,u], h'[3,u]] = \\ & = [h[0,u], h[1,u], h[2,u], h[3,u]] \oplus [hn[0,u], hn[1,u], hn[2,u], hn[3,u]], \end{aligned}$$

при $0 \leq u < 8$, где *hn* – элемент массива *holatn*, *h'* – элемент результирующего массива. Результат преобразования копируется в массив *holat*.

Если длина ключа хэширования и длина входного блока составляют 256 bit, тогда в преобразовании *BaytZichlash(holat, holatn)* каждый из массивов *holat*, *holatn* приводится в форму линейного массива *holat₁*, *holatn₁* на байтовом уровне и осуществляется замена байтов на полубайтовые элементы линейного массива сжатия порядка 1×256 и имеющий адресацию элементов от 0 до 255. Процедура замены заключается в замене каждого элемента линейных массивов *holat₁* и *holatn₁* на полубайтовый элемент линейного массива сжатия находящегося по адресу равному значению элемента линейных массивов *holat₁* и *holatn₁*.

В результате преобразования формируется пара линейных массивов *holat₂*, *holatn₂* порядка 1×256 с элементами на полубайтовом уровне. Преобразование завершается выполнением конкатенации $holat_3 = holat_2 \parallel holatn_2$ соответствующих элементов массивов и копированием элементов *holat₃* на байтовом уровне в массив *holat* порядка 4×8 .

6.2.3 Преобразование *Aralash (holat, k_o)*

Преобразование *Aralash(holat, k_o)* сводится к выполнению следующих действий.

1) Копировать двумерный массив *holat* в пару двумерных массивов *H_{ch}[4,4]* и *H_o[4,4]*.

2) Вычислить $H_{ch}' \equiv H_{ch} \otimes k_{ech} \pmod{p}$ и результат H_{ch}' присвоить массиву *H_{ch}*.

Для $s, u \in \{0, 1, 2, 3\}$ определить результат в форме:

$$h'_{ch}[u, u] \equiv h_{ch}[u, u] \times \sum_{i=0}^3 k_{ech}[i, u] - \sum_{i=0, i \neq u}^3 h_{ch}[i, i] \times k_{ech}[i, u] \pmod{p},$$

$$h'_{ch}[s, u]_{s \neq u} \equiv h_{ch}[s, u] \times \sum_{i=0}^3 k_{ech}[i, u] + k_{ech}[s, u] \times \sum_{i=0}^3 h_{ch}[i, u] - \sum_{i=0, i \neq s, u}^3 h_{ch}[s, i] \times k_{ech}[i, u] \pmod{p},$$

где: $h_{ch}[s, u]$ – элемент H_{ch} ; $k_{ech}[s, u]$ – элемент k_{ech} , который относится к левой половине массива $k_e[4, 8]$.

3) Вычислить $H_o' \equiv H_o \otimes k_{eo} \pmod{p}$ и результат H_o' присвоить в массив H_o .

Для $s, u \in \{0, 1, 2, 3\}$ определить результат в форме:

$$h'_o[u, u] \equiv h_o[u, u] \times \sum_{i=0}^3 k_{eo}[i, u] - \sum_{i=0, i \neq u}^3 h_o[i, i] \times k_{eo}[i, u] \pmod{p},$$

$$h'_o[s, u]_{s \neq u} \equiv h_o[s, u] \times \sum_{i=0}^3 k_{eo}[i, u] + k_{eo}[s, u] \times \sum_{i=0}^3 h_o[i, u] - \sum_{i=0, i \neq s, u}^3 h_o[s, i] \times k_{eo}[i, u] \pmod{p},$$

где: $h_o[s, u]$ – элемент H_o ; $k_{eo}[s, u]$ – элемент k_{eo} , который относится к правой половине массива $k_e[4, 8]$.

4) Массивы H_{ch} и H_o копировать в массив *holat*.

6.2.4 Преобразование *SurHolat(holat)*

0 строку массива *holat* циклически сдвинуть вправо на 1 полубайт (байт), первую строку на 2 полубайта (байта), вторую строку на 3 полубайта (байта), третью строку на 4 полубайта (байта), а 0 и 4 столбец полученного массива циклически сдвинуть вниз на 1 полубайт (байт), первый и пятый столбец на 2 полубайта (байта), второй и шестой столбцы на 3 полубайта (байта).

6.2.5 Преобразование *Sur Kalit(k_e)*

0 строку массива k_e циклически сдвинуть вправо на 4 полубайта (байта), первую строку на 3 полубайта (байта), вторую строку на 2 полубайта (байта), третью строку на 1 полубайт (байт), а первый и пятый столбец полученного массива циклически сдвинуть вверх на 3 полубайта (байта), второй и шестой столбцы на 2 полубайта (байта), третий и седьмой столбец на 1 полубайт (байт).

6.2.6 Преобразование *TuzilmaKalit(k_e, k)*

Преобразование *TuzilmaKalit(k_e, k)* осуществляется над каждым полубайтом (байтом) массива k_e с целью приведения её структуры к структуре исходного ключа хэширования k .

Для $0 \leq s < 4$, $0 \leq u < 8$:

- если $k_e[s, u] \equiv 0 \pmod{p}$ и $k[s, u] \equiv 0 \pmod{2}$, то принимается $k_e[s, u] = 2$;

- если $k_e[s, u] \not\equiv 0 \pmod{p}$ и $k_e[s, u] \pmod{2} \equiv k[s, u] \pmod{2}$, то принимается $k_e[s, u] = k_e[s, u]$, иначе принимается $k_e[s, u] \equiv k_e[s, u] - 1 \pmod{p}$.

Результат данного преобразования удовлетворяет условиям обратимости каждой из квадратных частей массива k_e .

В приложении А приведен пример выполнения процедуры хэширования, где показаны значения элементов массивов *holat* и k_e в начале сеанса и после применения каждого из упомянутых преобразований.

В приложении В приведен линейный массив *SurHolat* и *Sur Kalit* в порядке 1x256.

7 Алгоритм 2. Основные процессы

7.1 Шаговая функция хэширования

Алгоритм 2 вычисления шаговой функции хэширования включает в себя три части, реализующие последовательно:

- генерацию ключей – слов длиной 256 битов;
- шифрующее преобразование – зашифрование 64-битных подслов слова H на ключах K_i ($i=1,2,3,4$) с использованием алгоритма по ГОСТ 28147 в режиме простой замены;
- перемешивающее преобразование результата шифрования.

7.1.1 Генерация ключей

Рассмотрим вектор $X = (b_{256}, b_{255}, \dots, b_1) \in V_{256}(2)$.

Пусть $X = x_4 || x_3 || x_2 || x_1 = \eta_{16} || \eta_{15} || \dots || \eta_1 =$
 $= \xi_{32} || \xi_{31} || \dots || \xi_1,$

где: $x_i = (b_{ix64}, \dots, b_{(i-1)x64+1}) \in V_{64}(2), i = \overline{1,4};$

$\eta_j = (b_{jx16}, \dots, b_{(j-1)x16+1}) \in V_{16}(2), j = \overline{1,16};$

$\xi_k = (b_{kx8}, \dots, b_{(k-1)x8+1}) \in V_8(2), k = \overline{1,32}.$

Обозначают $A(X) = (x_1 \oplus x_2) || x_4 || x_3 || x_2.$

Используют преобразование $P: V_{256}(2) \rightarrow V_{256}(2)$ для отображения слова $\xi_{32} || \dots || \xi_1$ в слово $\xi_{\varphi(32)} || \dots || \xi_{\varphi(1)}$, где $\varphi(i+1+4(k-1)) = 8i+k, i = 0 \div 3, k = 1 \div 8.$

Для генерации ключей необходимо использовать следующие исходные данные:

- слова $H, M \in V_{256}(2);$
- параметры: слова C_i ($i=2,3,4$), имеющие значения $C_2=C_4=0^{256}$ и $C_3=1^8 0^8 1^{16} 0^{24} 1^{16} 0^8 (0^8 1^8)^2 1^8 0^8 (0^8 1^8)^4 (1^8 0^8)^4.$

При вычислении ключей реализуется следующий алгоритм:

- 1) присвоить значения
 $i := 1, U := H, V := M;$
- 2) выполнить вычисление
 $W = U \oplus V, K_i = P(W);$
- 3) присвоить $i := i+1;$
- 4) проверить условие. $i = 5.$

При положительном исходе перейти к шагу 7. При отрицательном – перейти к шагу 5;

- 5) выполнить вычисление:

$$U := A(U) \oplus C_b, V := A(A(V)), W := U \oplus V, K_i = P(W);$$

- 6) перейти к шагу 3;
- 7) конец работы алгоритма.

7.1.2 Шифрующее преобразование

На данном этапе осуществляется зашифрование 64-битных подслов слова H на ключах K_i ($i=1,2,3,4$).

Для шифрующего преобразования необходимо использовать следующие исходные данные:

$$- H = h_4 \| h_3 \| h_2 \| h_1, h_i \in V_{64}(2), i = \overline{1,4};$$

- набор ключей K_1, K_2, K_3, K_4 .

Реализуют алгоритм зашифрования и получают слова:

$$s_i = E_{K_i}(h_i), (i=1,2,3,4).$$

В результате данного этапа образуется последовательность

$$S = s_4 \| s_3 \| s_2 \| s_1.$$

7.1.3 Перемешивающее преобразование

На данном этапе осуществляется перемешивание полученной последовательности с применением регистра сдвига.

Исходными данными являются:

- слова $H, M \in V_{256}(2)$;

- слово $S \in V_{256}(2)$.

Пусть отображение

$$\psi: V_{256}(2) \longrightarrow V_{256}(2)$$

преобразует слово

$$\eta_{16} \| \eta_{15} \| \dots \| \eta_1, \eta_i \in V_{16}(2), i = \overline{1,16}$$

в слово

$$\eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_4 \oplus \eta_{13} \oplus \eta_{16} \| \eta_{16} \| \eta_{15} \| \dots \| \eta_2.$$

Тогда в качестве значения шаговой функции хэширования принимается слово

$$x(M, H) = \psi^{61}(H \oplus \psi(M \oplus \psi^{12}(S))),$$

где: ψ^i - i -я степень преобразования ψ .

7.2 Процедура вычисления хэш-функции

Исходными данными для процедуры вычисления значения функции хэширования h является подлежащая хэшированию последовательность $M \in B^*$. Параметром является стартовый вектор хэширования H - произвольное фиксированное слово из $V_{256}(2)$.

Процедура вычисления функции h на каждой итерации использует следующие величины:

$M \in B^*$ - часть последовательности M , не прошедшая процедуры хэширования на предыдущих итерациях;

$H \in V_{256}(2)$ - текущее значение хэш-функции;

$\Sigma \in V_{256}(2)$ - текущее значение контрольной суммы;

$L \in V_{256}(2)$ - текущее значение длины, обработанной на предыдущих итерациях части последовательности M .

Алгоритм вычисления функции h включает в себя этапы:

Этап 1

Присвоить начальные значения текущих величин

1.1 $M := M$

1.2 $H := H$

1.3 $\Sigma := 0^{256}$

1.4 $L := 0^{256}$

1.5 Переход к этапу 2.

Этап 2

2.1 Проверить условие $|M| > 256$. При положительном исходе перейти к этапу 3, в противном случае выполнить последовательность вычислений:

2.2 $L := \langle L + |M| \rangle_{256}$

2.3 $M' := 0^{256 - |M|} || M$

2.4 $\Sigma := \Sigma \oplus M'$

2.5 $H := x(M', H)$

2.6 $H := x(L, H)$

2.7 $H := x(\Sigma, H)$

2.8 Конец работы алгоритма.

Этап 3

3.1 Вычислить подслово $M_s \in V_{256}(2)$ слова $M (M = M_p || M_s)$. Далее выполнить последовательность вычислений:

3.2 $H := x(M_s, H)$

3.3 $L := \langle L + 256 \rangle_{256}$

3.4 $\Sigma := \Sigma \oplus M_s$

3.5 $M := M_p$

3.6 Перейти к этапу 2.

Значение величины H , полученное на шаге 2.7, является значением функции хэширования $h(M)$.

Проверочные примеры для вышеизложенной процедуры вычисления хэш-функции приведены в приложении С.

Приложение А

(справочное)

Контрольный пример для алгоритма 1

В приведенном приложении все числа представлены в шестнадцатеричной системе счисления.

Хэшируемое сообщение – исходные данные имеет следующее значение:

M= funksiyaxeshirovaniyainformasionnayatekhnologiyakriptograficheska

Число блоков **b=2**

Число этапов **e=10**

Ключ хэширования имеет следующее значение:

**к = c8 6e d4 b2 75 6d f1 cd 6b 6e ca 81 d4 41 80 92 21 ab 81 bb a0 c7
c1 50 44 be 8f eb 93 4f 95 a2**

Модуль **p=256**

Линейный массив сжатия приведен в приложении В.

ПРОЦЕСС ХЭШИРОВАНИЯ

После выполнения шагов по псевдокоду хэширования были получены следующие результаты:

i = 1

Qo'sh(holat, holatn):

holat=

8a	ff	7e	0d	b7	8b	49	25
f8	25	a1	68	43	12	91	be
9d	be	a9	d1	bf	37	d2	f6
8f	d2	73	a1	5d	9f	c5	5e

holatn=

8a	17	b1	fb	24	c3	f8	86
df	ec	11	07	17	e9	b3	af
4a	01	70	d4	0d	35	46	b9
e6	b9	49	68	4f	11	c5	29

BaytZichlash(holat, holatn)=

44	a6	2b	c4	3c	65	d7	7a
79	7d	c7	b8	26	2a	8d	c7
df	cd	8b	5b	5c	4d	49	cf
24	4f	1d	cb	f6	17	88	d9

j = 1

Aralash(holat, ke)=

4a	9f	4a	b9	0d	04	84	65
91	f6	ca	1b	13	e6	82	5b
4d	fa	23	1e	6f	cc	f6	b9
9e	36	96	45	78	23	7b	d9

U'ZBISTAT AGENTLIGI
STAFKARTLASHTIRISH,
NOYOFIDLASHTIRISH
VA DAVLAT NAZDRATI
BOSQIQLARI

Qo'sh(holat, holatn):**holat=**

56	eb	86	59	d3	84	24	1b
ab	c2	fe	c9	c3	fe	5e	87
5b	da	f1	4e	3f	3c	8e	ed
82	42	96	ff	d8	fd	91	d9

ke=holatn=

a8	9a	64	2e	dd	37	cb	35
a7	ae	6a	fd	cc	6f	80	e6
f1	49	8d	d1	60	21	0f	f0
f4	be	d5	ff	e5	7f	79	fe

SurHolat(holat)=

d8	8e	ab	86	82	f1	c3	24
1b	fd	ed	c2	59	42	4e	fe
5e	56	91	5b	fe	d3	96	3f
3c	87	eb	d9	da	c9	84	ff

SurKalit(ke)=

dd	f4	f1	a7	a8	e5	60	cc
6f	37	be	49	ae	9a	7f	21
0f	80	cb	d5	8d	6a	64	79
fe	f0	e6	35	ff	d1	fd	2e

Qo'sh(holat, holatn):**holat=**

b8	e2	d9	ce	6e	ff	f7	4c
0d	df	d1	42	35	e6	b6	2e
86	aa	6b	a1	1e	73	fa	1d
64	b3	63	47	c2	8f	b4	ef

ke=holatn=

27	34	59	55	28	89	a0	9c
1b	f7	62	2f	c2	b2	1d	ef
0d	80	6b	45	2b	1e	64	49
ae	90	02	29	95	8d	8f	6a

TuzilmaKalit(ke, k)=

26	34	58	54	27	89	9f	9b
1b	f6	62	2f	c2	b1	1c	ee
0d	7f	6b	45	2a	1d	63	48
ae	90	01	29	95	8d	8f	6a

- - - - -
- - - - -
- - - - -
- - - - -

j = 8**Aralash(holat, ke)=**

02	56	14	96	49	91	ae	d4
6b	5e	27	f1	52	aa	e8	86
0c	da	da	26	e5	78	54	c9
90	0a	5e	f0	b2	5b	5c	ea

Qo'sh(holat, holatn):

holat=

6a	fe	34	de	65	81	a2	4c
77	ee	cd	91	62	9e	08	96
94	8e	76	82	bf	48	2c	db
d0	42	62	50	c6	17	d4	0a

ke=holatn=

fc	74	a8	dc	dd	63	2b	c5
b7	16	28	8b	5a	b1	d6	de
45	49	d7	21	5e	81	09	b2
66	88	17	c7	25	ff	17	4e

SurHolat(holat)=

c6	2c	77	34	d0	76	62	a2
4c	17	db	ee	de	42	82	9e
08	6a	d4	94	cd	65	62	bf
48	96	fe	0a	8e	91	81	50

SurKalit(ke)=

dd	66	45	b7	fc	25	5e	5a
b1	63	88	49	16	74	ff	81
09	d6	2b	17	d7	28	a8	17
4e	b2	de	c5	c7	21	8b	dc

Qo'sh(holat, holatn):

holat=

86	24	03	bc	50	8a	4a	d2
b4	09	e3	5a	52	3e	5e	ba
e8	36	34	d4	61	59	82	eb
e8	ba	3a	f2	d2	15	d5	f0

holatn=

96	e3	59	53	04	13	78	e6
a9	d4	9b	a5	b7	bf	87	19
02	1f	30	a4	df	d7	2a	c1
a6	c7	71	e8	a3	03	ab	1b

BaytZichlash(holat, holatn)=

a2	c6	bb	e8	21	4e	f3	44
78	4b	6f	61	13	a5	dc	ac
26	70	30	b2	49	bd	30	b1
2e	a7	59	52	40	3b	2d	da

TuzilmaKalit(ke, k)=

dc	66	44	b6	fb	25	5d	59
b1	62	88	49	16	73	fe	80
09	d5	2b	17	d6	27	a7	16
4e	b2	dd	c5	c7	21	8b	dc

j = 9

Aralash(holat, ke)=

c1	23	f0	f3	aa	6e	83	cd
79	a5	67	a7	8c	31	89	61
0a	f0	16	3d	c7	cf	5f	0c
e4	f8	4b	47	bb	d0	e3	c5

Q'ZSTATISTIK ADO'ZLASH
STATISTIKALASHTIRISH
NOYOFIDOLASHITIRISH
VA BAYLAT NAZORATI
DO'ZSTATISTIK

Qo'sh(holat, holatn):

holat=

c5	af	50	3f	3a	c6	c1	53
1f	d5	3f	cd	94	63	c9	dd
72	f0	76	87	fb	f1	01	14
fc	c8	29	05	dd	50	cd	61

ke=holatn=

94	82	04	72	b9	73	51	c9
89	7e	78	fd	de	bf	a2	80
ef	87	51	03	62	cf	af	de
b6	4a	21	f1	37	83	33	34

SurHolat(holat)=

dd	01	1f	50	fc	76	94	c1
53	50	14	d5	3f	c8	87	63
c9	c5	cd	72	3f	3a	29	fb
f1	dd	af	61	f0	cd	c6	05

SurKalit(ke)=

b9	b6	ef	89	94	37	62	de
bf	73	4a	87	7e	82	83	cf
af	a2	51	21	51	78	04	33
34	de	80	c9	f1	03	fd	72

Qo'sh(holat, holatn):

holat=

73	01	a1	d0	f4	46	6c	71
75	10	2c	ef	73	78	51	25
73	b5	b3	da	85	f6	6d	45
45	65	87	47	70	4f	5e	55

ke=holatn=

35	d2	eb	4f	84	29	b2	2a
33	61	7a	57	42	42	e7	83
27	f6	9d	eb	c1	38	fc	2f
0c	5a	80	4d	77	7b	df	a6

TuzilmaKalit(ke, k)=

34	d2	ea	4e	83	29	b1	29
33	60	7a	57	42	41	e6	82
27	f5	9d	eb	c0	37	fb	2e
0c	5a	7f	4d	77	7b	df	a6

Qo'sh(holat, holatn):

holat=

4b	bb	ab	10	0c	e6	34	e5
33	70	cc	0d	15	58	0b	4b
53	7f	5f	9e	03	ee	e1	5f
01	2f	9d	0b	b0	e5	c6	6f

holatn=

0e	b7	ef	fd	dc	ad	a8	d6
5d	a4	3f	89	4b	55	17	89
02	f7	b0	8c	fd	7b	aa	6f
7e	97	35	88	b9	d5	d7	4f

BaytZichlash(holat, holatn)=

83	23	d3	8b	21	43	35	06
cf	b2	68	ce	38	ce	96	8e
86	c0	46	0f	bb	10	fb	4d
d2	f5	dd	99	6f	02	3d	d6

j = 10

Aralash(holat, ke)=

82	0a	c9	b7	a6	aa	0c	bd
e4	74	64	a3	64	80	7e	74
8e	a0	87	ca	69	53	7c	02
d2	34	6c	6b	e5	27	ff	2b

Qo'sh(holat, holatn):

holat=

f2	62	93	09	5a	ee	d4	21
ec	84	e4	cf	a4	80	86	34
22	60	ff	86	4f	df	34	6a
a2	34	a4	8b	f5	63	63	1d

ke=holatn=

5c	9e	f2	4e	3f	89	45	c3
53	a0	0e	e5	c6	3d	0a	46
4b	d9	ab	af	c0	65	7b	72
a4	ee	e7	e3	71	e1	35	16

SurHolat(holat)=

f5	34	ec	93	a2	ff	a4	d4
21	63	6a	84	09	34	86	80
86	f2	63	22	e4	5a	a4	4f
df	34	62	1d	60	cf	ee	8b

SurKalit(ke)=

3f	a4	4b	53	5c	71	c0	c6
3d	89	ee	d9	a0	9e	e1	65
7b	0a	45	e7	ab	0e	f2	35
16	72	46	c3	e3	af	e5	4e

Qo'sh(holat, holatn):

holat=

29	14	b4	73	ca	cb	a4	a4
69	f7	aa	ec	1b	c4	ea	80
32	22	1f	26	3c	da	d4	73
25	a4	a2	69	a0	4f	32	a1

ke=holatn=

a1	fc	0b	59	b4	07	40	32
b7	67	c2	dd	a0	9a	89	99
17	36	43	73	3b	e2	46	ab
d6	86	6a	05	6b	5d	d5	de

TuzilmaKalit(ke, k)=

a0	fc	0a	58	b3	07	3f	31
b7	66	c2	dd	a0	99	88	98
17	35	43	73	3a	e1	45	aa
d6	86	69	05	6b	5d	d5	de

i = 2

Qo'sh(holat, holatn):

0'ZOTAT'AST ADEITLIGI
STAN'ITLASHTIRISH,
NOLIFIRLASHTIRISH
VA DAYLAT RAZDRATI
BOSONARHAGI

holat=

7b	a4	c4	0d	3a	65	f4	f4
bb	b1	9a	1c	b5	94	da	80
42	f2	59	26	2c	8a	64	0d
f7	f4	72	bb	20	89	0a	b3

holatn=

08	86	c6	2c	12	1c	a6	a6
88	14	d2	b6	3c	e6	52	9e
62	42	24	8a	56	32	06	2c
20	a6	42	88	de	04	a2	38

BaytZichlash(holat, holatn)=

00	2a	a3	c3	52	55	8e	8e
29	bf	c4	50	9b	e4	31	10
7e	57	bc	14	35	49	9f	c3
0e	8e	a7	29	ef	e1	7f	df

j = 1

Aralash(holat, ke)=

4d	b9	7d	cc	b5	07	e3	6a
cb	41	8a	fc	fe	b8	04	db
d2	01	91	09	11	07	e6	b6
83	8f	fa	5e	3d	ed	27	e7

Qo'sh(holat, holatn):

holat=

59	95	1d	d4	2b	2d	d9	02
91	11	86	1c	da	f8	9c	5b
72	cb	e7	ff	69	b9	2e	ca
67	7b	32	e6	a3	57	d1	f3

ke=holatn=

60	84	9e	98	5f	cf	4f	5b
67	32	22	73	20	27	c8	08
a9	c5	e7	27	fe	1d	57	da
ba	5a	73	a7	17	a5	29	42

SurHolat(holat)=

a3	2e	91	1d	67	e7	da	d9
02	57	ca	11	d4	7b	ff	f8
9c	59	d1	72	86	2b	32	69
b9	5b	95	f3	cb	1c	2d	e6

SurKalit(ke)=

5f	ba	a9	67	60	17	fe	20
27	cf	5a	c5	32	84	a5	1d
57	c8	4f	73	e7	22	9e	29
42	da	08	5b	a7	27	73	98

Qo'sh(holat, holatn):

holat=

11	52	0b	5b	db	75	e6	69
9a	d5	a6	53	1c	df	69	18
3c	19	0f	7a	9e	3b	ee	63
35	7b	2d	59	69	3c	73	e2

ke=holatn=

b7	7a	a9	5b	a0	f7	9e	60
0d	3f	ea	8d	da	fc	89	37
89	18	2b	d9	65	e6	1e	d9
ee	ee	18	4b	3f	19	6f	38

TuzilmaKalit(ke, k)=

b6	7a	a8	5a	9f	f7	9d	5f
0d	3e	ea	8d	da	fb	88	36
89	17	2b	d9	64	e5	1d	d8
ee	ee	17	4b	3f	19	6f	38

- - - - -
 - - - - -
 - - - - -
 - - - - -

j = 8

Aralash(holat, ke)=

9c	a0	bf	27	08	48	c5	95
de	d6	24	60	a2	7f	87	24
4f	02	71	c0	c9	7b	0d	93
dc	36	45	6d	50	36	b9	db

Qo'sh(holat, holatn):

holat=

84	60	9f	27	e8	e8	6b	6b
36	6a	ac	e0	f6	01	1f	fc
85	7a	d3	c0	95	d5	17	13
34	62	97	7f	10	ae	97	63

ke=holatn=

f4	84	d2	1a	63	0f	f7	af
19	76	ba	6f	b0	6d	1e	d4
b3	d9	e5	dd	46	b9	71	ae
a6	b8	95	45	7b	f1	a3	22

SurHolat(holat)=

10	17	36	9f	34	d3	f6	6b
6b	ae	13	6a	27	62	c0	01
1f	84	97	85	ac	e8	97	95
d5	fc	60	63	7a	e0	e8	7f

SurKalit(ke)=

63	a6	b3	19	f4	7b	46	b0
6d	0f	b8	d9	76	84	f1	b9
71	1e	f7	95	e5	ba	d2	a3
22	ae	d4	af	45	dd	6f	1a

Qo'sh(holat, holatn):

holat=

b0	8b	2a	03	5c	6f	6a	57
57	12	2f	a6	7b	8e	40	79
83	cc	0b	75	04	f8	0b	e5
a5	74	20	df	d6	a0	d8	23

0'ZOTAPART AGENTLIGI
 STAN. QILASHTIRISH,
 HAJJATLASHIRISH
 VA DAYLAT NAZDRATI
 BOSHQARUVCHI

holatn=

66	80	1a	30	1e	28	9a	38
38	2a	a8	32	e0	42	06	3c
30	7e	80	14	2e	b6	80	74
f4	8e	c6	88	12	c6	82	70

BaytZichlash(holat, holatn)=

68	61	0d	b0	a9	d6	1c	9f
9f	20	f5	e9	07	57	3f	5b
70	62	91	4f	18	70	91	06
18	65	e3	99	62	93	83	4b

TuzilmaKalit(ke, k)=

62	a6	b2	18	f3	7b	45	af
6d	0e	b8	d9	76	83	f0	b8
71	1d	f7	95	e4	b9	d1	a2
22	ae	d3	af	45	dd	6f	1a

j = 9**Aralash(holat, ke)=**

dd	b6	e8	be	ed	49	26	7a
aa	05	fb	15	b7	91	83	b2
c7	74	09	de	da	ba	a3	60
fa	c4	16	68	c8	ab	c6	70

Qo'sh(holat, holatn):**holat=**

c1	ea	f8	da	2b	47	a6	92
7a	01	17	f7	17	df	5f	06
31	04	ef	fe	ce	ba	cd	20
e6	ac	de	e8	a8	0d	5e	d0

ke=holatn=

ce	26	7a	f8	8f	8f	fb	95
cb	92	c8	31	12	37	10	58
0d	cf	d3	7b	34	97	e5	0a
82	a6	81	19	df	99	9d	82

SurHolat(holat)=

a8	cd	7a	f8	e6	ef	17	a6
92	0d	20	01	da	ac	fe	df
5f	c1	5e	31	17	2b	de	ce
ba	06	ea	d0	04	f7	47	e8

SurKalit(ke)=

8f	82	0d	cb	ce	df	34	12
37	8f	a6	cf	92	26	99	97
e5	10	fb	81	d3	c8	7a	9d
82	0a	58	95	19	7b	31	f8

Qo'sh(holat, holatn):**holat=**

a8	61	aa	38	2a	7d	53	5a
4a	2b	60	6f	76	04	6e	5d
39	21	c6	4b	cd	17	a2	2e
d6	9a	5e	90	b4	3d	59	98

ke=holatn=

69	6e	7b	f5	6e	0f	4c	02
9d	83	ce	fb	b2	3e	ef	97
29	30	49	31	5b	b8	ea	13
e2	1a	f8	ef	8b	d3	3d	f8

TuzilmaKalit(ke, k)=

68	6e	7a	f4	6d	0f	4b	01
9d	82	ce	fb	b2	3d	ee	96
29	2f	49	31	5a	b7	e9	12
e2	1a	f7	ef	8b	d3	3d	f8

Qo'sh(holat, holatn):

holat=

98	3d	de	88	5e	b1	cb	2e
3e	93	a0	df	32	bc	ba	51
c5	7d	e2	f3	a1	17	e6	fa
d2	ee	ca	f0	8c	f1	e9	28

holatn=

64	1a	20	44	20	82	c6	c0
e0	56	f4	2e	08	ac	18	c2
aa	9a	68	16	62	9e	30	98
48	40	38	94	4c	02	32	84

BaytZichlash(holat, holatn)=

a9	2d	fe	94	de	b3	c3	8f
a7	95	98	98	90	e4	a1	30
8b	7c	cb	3b	ce	60	40	2a
4e	13	9f	de	f0	66	a9	6d

j = 10

Aralash(holat, ke)=

b0	29	ee	b3	41	d3	6f	ad
0d	be	82	4b	2e	ff	b5	b2
1f	c3	36	9a	22	0b	c5	68
f4	c0	02	fc	4e	0c	ec	97

Qo'sh(holat, holatn):

holat=

10	cd	0a	97	b7	ad	61	2b
43	5a	e6	cd	2a	5d	d1	16
a5	1d	56	6a	d6	b5	d3	98
ac	40	9a	3c	56	6c	1c	0b

ke=holatn=

d8	ce	d6	d4	2f	9d	21	2b
67	ee	82	49	be	a3	de	9a
9f	9d	b5	c5	c6	15	63	06
26	5e	77	33	7b	7f	d5	38

SurHolat(holat)=

56	d3	43	0a	ac	56	2a	61
2b	6c	98	5a	97	40	6a	5d
d1	10	1c	a5	e6	b7	9a	d6
b5	16	cd	0b	1d	cd	ad	3c

U'ZBEKISTAN AGENTLIGI
 STATISTIKALASHTIRISH,
 MONITORING VA
 VA BAYLAT NAZORATI
 QURULMASI

SurKalit(ke)=

2f	26	9f	67	d8	7b	c6	be
a3	9d	5e	9d	ee	ce	7f	15
63	de	21	77	b5	82	d6	d5
38	06	9a	2b	33	c5	49	d4

Qo'sh(holat, holatn):**holat=**

9e	4b	29	92	f4	2e	96	e5
19	3c	08	ba	57	c0	d2	37
4f	b0	0c	8b	56	1f	26	06
6d	fa	e1	49	6b	47	5f	04

ke=holatn=

83	b6	f5	03	28	d7	2a	46
29	39	8a	55	2e	ba	6b	f7
ed	ea	65	09	75	12	9a	b5
78	5a	32	31	65	47	db	cc

TuzilmaKalit(ke, k)=

82	b6	f4	02	27	d7	29	45
29	38	8a	55	2e	b9	6a	f6
ed	e9	65	09	74	11	99	b4
78	5a	31	31	65	47	db	cc

i = 3**Qo'sh(holat, holatn):****holat=**

42	93	ab	de	74	da	56	e7
db	0c	78	b2	7f	c0	ce	65
49	30	5c	21	5a	8b	8a	3e
8b	42	d1	4f	2f	4b	07	c4

holatn=

7c	de	69	84	23	e2	35	e9
c9	d9	26	07	aa	a7	70	df
b1	cb	91	d3	f0	d0	a0	47
f7	db	10	73	98	7c	92	fb

BaytZichlash(holat, holatn)=

78	9f	d0	fd	64	3c	5d	ea
e2	27	31	88	cb	fa	4b	59
db	0c	a8	2a	6d	6c	49	a1
60	7e	58	61	fa	88	8b	a4

j = 1**Aralash(holat, ke)=**

41	0d	11	52	44	93	61	8a
e0	25	ee	db	55	3f	bf	40
05	a6	29	a8	a1	78	2b	12
32	e6	a5	26	ae	b5	87	48

Qo'sh(holat, holatn):**holat=**

31	cd	25	7e	74	29	e3	42
e0	89	12	15	f5	61	cb	c0
bf	0e	e3	a8	d5	f8	7d	76
86	3a	e7	ae	86	b3	b5	f8

ke=holatn=

16	12	0c	e2	25	af	79	07
9f	48	fa	99	ba	ed	a6	2e
7d	33	dd	af	0c	f7	95	a4
58	6a	69	3b	5f	73	b3	2c

SurHolat(holat)=

86	7d	e0	25	86	e3	f5	e3
42	b3	76	89	7e	3a	a8	61
cb	31	b5	bf	12	74	e7	d5
f8	c0	cd	f8	0e	15	29	ae

SurKalit(ke)=

25	58	7d	9f	16	5f	0c	ba
ed	af	6a	33	48	12	73	f7
95	a6	79	69	dd	fa	0c	b3
2c	a4	2e	07	3b	af	99	e2

Qo'sh(holat, holatn):

holat=

d6	0d	60	df	a2	ad	7d	5b
c2	9d	82	1b	f2	1e	a8	3b
a9	75	d3	75	12	6c	93	37
28	40	79	38	56	2f	47	ea

ke=holatn=

ef	a8	a3	6b	56	f7	14	66
7f	c7	da	8f	a8	32	c9	eb
e9	12	31	ed	cf	22	44	07
4c	84	9a	ad	7d	fb	49	92

TuzilmaKalit(ke, k)=

ee	a8	a2	6a	55	f7	13	65
7f	c6	da	8f	a8	31	c8	ea
e9	11	31	ed	ce	21	43	06
4c	84	99	ad	7d	fb	49	92

- - - - -
 - - - - -
 - - - - -
 - - - - -

j = 8

Aralash(holat, ke)=

bd	f8	b7	c3	22	3f	72	bc
e4	6d	c3	c8	ad	c9	e6	f4
ae	3c	6f	66	67	c8	a1	97
a4	76	c2	68	cd	fd	86	1e

Qo'sh(holat, holatn):

holat=

15	c8	3b	93	b2	d1	aa	ec
54	dd	17	a8	05	73	e2	7c
66	5c	5d	c6	43	88	97	13
0c	ba	d2	08	ff	3b	1e	22

0'ZS'YAC'AST AGENTLIGI
 STAN QATLACHTIRISH
 NUQDIDLASHTIRISH
 VA DAVLAT NAZORATI
 BOSONAHAB

ke=holatn=

74	08	6c	96	9d	47	d3	29
6d	f8	e8	51	e4	63	14	4a
fb	f7	a5	6d	18	23	ed	38
5a	fe	9d	5d	47	89	93	4a

SurHolat(holat)=

ff	97	54	3b	0c	5d	05	aa
ec	3b	13	dd	93	ba	c6	73
e2	15	1e	66	17	b2	d2	43
88	7c	c8	22	5c	a8	d1	08

SurKalit(ke)=

9d	5a	fb	6d	74	47	18	e4
63	47	fe	f7	f8	08	89	23
ed	14	d3	9d	a5	e8	6c	93
4a	38	4a	29	5d	6d	51	96

Qo'sh(holat, holatn):**holat=**

09	d5	4c	b5	94	eb	e1	5e
b4	43	39	7d	19	fe	46	ab
76	15	a2	2a	59	02	e2	c7
d8	04	a8	ee	4c	c8	d7	f8

holatn=

ac	82	39	14	29	12	65	9d
e9	79	86	4f	16	09	d0	f9
e1	b7	6f	c3	b0	50	e0	73
57	dd	10	37	78	2c	c6	c9

BaytZichlash(holat, holatn)=

44	23	06	9f	e9	b2	f5	dd
7a	25	6a	76	cb	e4	9c	d1
df	33	fd	05	b6	62	c7	71
89	10	58	14	03	d3	d3	72

TuzilmaKalit(ke, k)=

9c	5a	fa	6c	73	47	17	e3
63	46	fe	f7	f8	07	88	22
ed	13	d3	9d	a4	e7	6b	92
4a	38	49	29	5d	6d	51	96

j = 9**Aralash(holat, ke)=**

08	c5	c9	6c	86	c5	77	0e
30	e8	f4	ce	5f	8e	ee	06
0d	a4	f2	ba	9f	9a	9c	29
3a	66	63	a8	49	bb	3f	a9

Qo'sh(holat, holatn):**holat=**

48	6b	4f	2c	02	c9	c7	ba
50	88	64	8a	61	0a	b6	56
3d	9c	f6	26	09	fe	a4	1f
1a	ae	a7	38	41	a7	5b	47

ke=holatn=

c4	c2	12	14	37	01	cd	37
a3	52	0a	e3	18	d3	98	9e
c7	cb	bf	dd	f4	bb	a3	aa
46	28	af	7d	0f	43	2f	4e

SurHolat(holat)=

41	a4	50	4f	1a	f6	61	c7
ba	a7	1f	88	2c	ae	26	0a
b6	48	5b	3d	64	02	a7	09
fe	56	6b	47	9c	8a	c9	38

SurKalit(ke)=

37	46	c7	a3	c4	0f	f4	18
d3	01	28	cb	52	c2	43	bb
a3	98	cd	af	bf	0a	12	2f
4e	aa	9e	37	7d	dd	e3	14

Qo'sh(holat, holatn):

holat=

37	2c	10	b5	be	06	8d	ff
2a	61	a7	28	64	e2	c6	9a
96	78	cd	83	74	be	7b	2f
42	3a	ff	c5	7c	f2	f7	c8

ke=holatn=

3b	ae	f5	ab	34	a9	ac	08
15	15	38	99	fa	32	1d	dd
9d	98	21	fb	55	ea	ce	33
0e	da	c2	c7	e3	87	cf	34

TuzilmaKalit(ke, k)=

3a	ae	f4	aa	33	a9	ab	07
15	14	38	99	fa	31	1c	dc
9d	97	21	fb	54	e9	cd	32
0e	da	c1	c7	e3	87	cf	34

Qo'sh(holat, holatn):

holat=

bf	c4	50	a1	3e	1a	37	bd
1a	bb	13	c8	cc	62	a2	22
a6	d8	87	75	cc	2a	d3	bd
3a	da	7f	bf	c4	a6	93	e8

holatn=

4c	d6	09	64	a5	46	25	33
5d	dd	66	21	82	a1	d0	c7
51	89	3d	7d	90	d0	20	bb
9b	81	30	dd	f8	5c	ee	61

BaytZichlash(holat, holatn)=

50	a6	24	c9	a1	d9	47	1c
df	20	e8	d2	63	ec	fc	d7
e3	8e	c2	47	64	0c	ae	12
5f	30	c0	50	a7	ea	91	24

U'ZBEKISTON RESPUBLIKASI
SANKTSLASHTIRISH
MUNOFIDLASHTIRISH
VA DAVLAT HAZDRATI
KUTUBXONASI

j = 10

Aralash(holat, ke)=

26	64	77	9f	68	ca	eb	c1
47	72	e2	6e	fd	95	b2	9e
73	66	0c	cd	4e	29	d3	18
44	26	82	1a	71	09	0c	11

Qo'sh(holat, holatn):

holat=

0e	84	e1	75	d8	66	5f	71
67	a2	22	12	b3	59	82	3e
4b	a2	44	2d	4e	d5	4b	d8
64	8e	2e	2e	89	49	d4	b3

ke=holatn=

c2	9e	0c	ae	5d	97	c7	c7
65	74	b8	bf	b6	cd	7c	dc
05	9d	bb	6b	14	2d	d5	82
fe	62	df	15	0b	17	81	1c

SurHolat(holat)=

89	4b	67	e1	64	44	b3	5f
71	49	d8	a2	75	8e	2d	59
82	0e	d4	4b	22	d8	2e	4e
d5	3e	84	b3	a2	12	66	2e

SurKalit(ke)=

5d	fe	05	65	c2	0b	14	b6
cd	97	62	9d	74	9e	17	2d
d5	7c	c7	df	bb	b8	0c	81
1c	82	dc	c7	15	6b	bf	ae

Qo'sh(holat, holatn):

holat=

55	39	4f	2d	84	fc	cd	4d
8d	01	d8	fe	f7	56	cd	65
7e	be	cc	3f	96	f8	1e	52
07	76	74	47	de	66	e2	b6

ke=holatn=

69	1a	d5	71	a2	65	6c	52
29	07	92	33	5c	66	4f	79
4b	bc	c1	33	31	58	cc	af
14	2a	6c	3b	4b	01	ad	f6

TuzilmaKalit(ke, k)=

68	1a	d4	70	a1	65	6b	51
29	06	92	33	5c	65	4e	78
4b	bb	c1	33	30	57	cb	ae
14	2a	6b	3b	4b	01	ad	f6

РЕЗУЛЬТАТ ХЭШИРОВАНИЯ:

55	39	4f	2d	84	fc	cd	4d
8d	01	d8	fe	f7	56	cd	65
7e	be	cc	3f	96	f8	1e	52
07	76	74	47	de	66	e2	b6

Приложение В

(обязательное)

Линейный массив сжатия порядка 1x256.

В приведенном приложении все числа представлены в шестнадцатеричной системе счисления.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
05	0d	06	0b	01	0a	0f	08	00	04	07	09	02	0c	03	0e
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
08	07	02	0e	0f	03	0b	06	01	0c	0d	0a	05	04	09	00
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0e	02	0d	04	0c	07	01	0b	06	09	00	05	03	0a	08	0f
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
00	0e	09	0c	03	0d	07	04	0f	06	05	01	0b	02	0a	08
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
03	0a	07	02	04	0c	09	01	0e	0d	0f	08	00	05	0b	06
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
02	03	01	08	00	0e	05	09	0c	0b	06	07	0a	0f	0d	04
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
0a	04	0e	0f	09	05	08	02	0b	00	01	03	0c	06	07	0d
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
0b	09	0a	01	06	04	0d	0f	03	05	0e	00	08	07	02	0c
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
01	00	03	07	0d	0b	0a	0c	09	0e	04	06	0f	08	05	02
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
04	08	0b	09	0e	06	02	05	0a	03	0c	0f	07	0d	00	01
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
09	0c	0f	00	02	01	0e	0a	05	08	0b	0d	04	03	06	07
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
06	0b	08	0d	07	09	00	03	04	0f	0a	02	0e	01	0c	05
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
0f	01	00	05	0a	08	03	07	0d	02	09	0c	06	0e	04	0b
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
0c	05	04	0a	0b	02	06	0d	08	07	03	0e	01	00	0f	09
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
07	0f	0c	06	05	00	04	0e	02	0a	08	0b	0d	09	01	03
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
0d	06	05	03	08	0f	0c	00	07	01	02	04	09	0b	0e	0a

Q'ZOTABDARI AGENTLARI
STAN QIRILASHTIRISH
KOLLEKTIVLASHTIRISH
VA BAYLAT NAZORATI
BO'LINMASI

Приложение С

(справочное)

Контрольный пример по алгоритму 2

Заполнение узлов замены $\pi_1, \pi_2, \dots, \pi_8$ и значение стартового вектора хэширования H , указанные в данном приложении, рекомендуется использовать только в проверочных примерах для настоящего стандарта.

С.1 Использование алгоритма ГОСТ 28147

В качестве шифрующего преобразования в приводимых ниже примерах используется алгоритм ГОСТ 28147 в режиме простой замены.

При этом заполнение узлов замены $\pi_1, \pi_2, \dots, \pi_8$ блока подстановки π следующее:

	8	7	6	5	4	3	2	1
0	1	D	4	6	7	5	E	4
1	F	B	B	C	D	8	B	A
2	D	4	A	7	A	1	4	9
3	0	1	0	1	1	D	C	2
4	5	3	7	5	0	A	6	D
5	7	F	2	F	8	3	D	8
6	A	5	1	D	9	4	F	0
7	4	9	D	8	F	2	A	E
8	9	0	3	4	E	E	2	6
9	2	A	6	A	4	F	3	B
10	3	E	8	9	6	C	8	1
11	E	7	5	E	C	7	1	C
12	6	6	9	0	B	6	0	7
13	B	8	C	3	2	0	7	F
14	8	2	F	B	5	9	5	5
15	C	C	E	2	3	B	9	3

В столбце с номером $j, j = \overline{0,18}$, в строке с номером $i, i = \overline{0,15}$, приведено значение $\pi_j(i)$ в шестнадцатеричной системе счисления.

С.2 Представление векторов

Последовательности двоичных символов будем записывать как строки шестнадцатеричных цифр, в которых каждая цифра соответствует четырем знакам ее двоичного представления.

С.3 Примеры вычисления значения хэш-функции

В качестве стартового вектора хэширования принимают, например, нулевой вектор

$H = 00000000\ 00000000\ 00000000\ 00000000$

00000000 00000000 00000000 00000000.

С.3.1 Пусть необходимо выполнить хэширование сообщения

$M = 73657479\ 62203233\ 3D687467\ 6E656C20$

$2C656761\ 7373656D\ 20736920\ 73696854.$

Выполняют присвоение начальных значений:

текста:

$M = 73657479\ 62203233\ 3D687467\ 6E656C20$

$2C656761\ 7373656D\ 20736920\ 73696854$

хэш-функции:

$H = 00000000\ 00000000\ 00000000\ 00000000$

$00000000\ 00000000\ 00000000\ 00000000$

суммы блоков текста:

$\Sigma = 00000000\ 00000000\ 00000000\ 00000000$

$00000000\ 00000000\ 00000000\ 00000000$

длины текста:

$L = 00000000\ 00000000\ 00000000\ 00000000$

$00000000\ 00000000\ 00000000\ 00000000.$

Так как длина сообщения, подлежащего хэшированию, равна 256 bit (32 byte),

$L = 00000000\ 00000000\ 00000000\ 00000000$

$00000000\ 00000000\ 00000000\ 00000100.$

$M' = M = 73657479\ 62203233\ 3D687467\ 6E656C20$

$2C656761\ 7373656D\ 20736920\ 73696854,$

нет необходимости дописывать текущий блок нулями,

$\Sigma = M' = 73657479\ 62203233\ 3D687467\ 6E656C20$

$2C656761\ 7373656D\ 20736920\ 73696854.$

Переходят к вычислению значения шаговой функции хэширования $x(M, H)$. Вырабатывают ключи:

$K_1 = 733D2C20\ 65686573\ 74746769\ 79676120$

$626E7373\ 20657369\ 326C6568\ 33206D54$

$K_2 = 110C733D\ 0D166568\ 130E7474\ 06417967$

$1D00626E\ 161A2065\ 090D326C\ 4D393320$

$K_3 = 80B111F3\ 730DF216\ 850013F1\ C7E1F941$

$620C1DFF\ 3ABAE91A\ 3FA109F2\ F513B239$

$K_4 = A0E2804E\ FF1B73F2\ ECE27A00\ E7B8C7E1$

$EE1D620C\ AC0CC5BA\ A804C05E\ A18B0AEC$

Осуществляют зашифрование 64-битных подслов блока H с помощью алгоритма по ГОСТ 28147

Блок $h_1 = 00000000\ 00000000$ зашифровывают на ключе K_1 и получают $s_1 = 42ABBCSE\ 32BC0B1B.$

STANDART AGENTLIGI
STANDARTLASHTIRISH
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
KOSHVANNASI

Блок $h_2 = 00000000\ 00000000$ зашифровывают на ключе K_2 и получают $s_2 = 5203EBC8\ 5D9BCFFD$.

Блок $h_3 = 00000000\ 00000000$ зашифровывают на ключе K_3 и получают $s_3 = 8D345899\ 00FF0E28$.

Блок $h_4 = 00000000\ 00000000$ зашифровывают на ключе K_4 и получают $s_4 = E7860419\ 0D2A562D$.

Получают:

$S = E7860419\ 0D2A562D\ 8D345899\ 00FF0E28$
 $5203EBC8\ 5D9BCFFD\ 42ABBCCE\ 32BC0B1B$.

Выполняют перемешивающее преобразование с применением регистра сдвига и получают:

$E = x(M, H) = CF9A8C65\ 505967A4\ 68A03B8C\ 42DE7624$
 $D99C4124\ 883DA687\ 561C7DE3\ 3315C034$.

Полагают $H = E$, вычисляют $x(L, H)$:

$K_1 = CF68D956\ 9AA09C1C\ 8C3B417D\ 658C24E3$
 $50428833\ 59DE3D15\ 6776A6C1\ A4248734$
 $K_2 = 8FCF68D9\ 809AA09C\ 3C8C3B41\ C7658C24$
 $BB504288\ 2859DE3D\ 666676A6\ B3A42487$
 $K_3 = 4E70CF97\ 3C8065A0\ 853C8CC4\ 57389A8C$
 $CABB50BD\ E3D7A6DE\ D1996788\ 5CB35B24$
 $K_4 = 584E70CF\ C53C8065\ 48853C8C\ 1657389A$
 $EDCABB50\ 78E3D7A6\ EED19867\ 7F5CB35B$
 $S = 66B70F5E\ F163F461\ 468A9528\ 61D60593$
 $E5EC8A37\ 3FD42279\ 3CD1602D\ DD783E86$
 $E = 2B6EC233\ C7BC89E4\ 2ABC2692\ 5FEA7285$
 $DD3848D1\ C6AC997A\ 24F74E2B\ 09A3AEF7$

Вновь полагают $H = E$ и вычисляют $x(\Sigma, H)$:

$K_1 = 5817F104\ 0BD45D84\ B6522F27\ 4AF5B00B$
 $A531B57A\ 9C8FD FCA\ BB1EFCC6\ D7A517A3$
 $K_2 = E82759E0\ C278D950\ 15CC523C\ FC72EBB6$
 $D2C73DA8\ 19A6CAC9\ 3E8440F5\ C0DDB65A$
 $K_3 = 77483AD9\ F7C29CAA\ EB06D1D7\ 841BCAD3$
 $FBC3DAA0\ 7CB555F0\ D4968080\ 0A9E56BC$
 $K_4 = A1157965\ 2D9FBC9C\ 008C7CC2\ 46FB3DD2$
 $7684ADCB\ FA4ACA06\ 53EFF7D7\ C0748708$
 $S = 2AEBFA76\ A85FB57D\ 6F164DE9\ 2951A581$
 $C31E7435\ 4930FD05\ 1F8A4942\ 550A582D$
 $E = FAFF37A6\ 15A81669\ 1CFF3EF8\ B68CA247$
 $E09525F3\ 9F811983\ 2EB81975\ D366C4B1$

Таким образом, результат хэширования есть:

$H = FAFF37A6\ 15A81669\ 1CFF3EF8\ B68CA247$
 $E09525F3\ 9F811983\ 2EB81975\ D366C4B1$

С.3.2 Пусть необходимо выполнить хэширование сообщения:

M= 7365 74796220 3035203D 20687467 6E656C20 73616820
 65676173 73656D20 6C616E69 6769726F 20656874 2065736F
 70707553

Так как длина сообщения, подлежащего хэшированию, равна 400 bit (50 byte), то разбивают сообщение на два блока и второй (старший) блок дописывают нулями. В процессе вычислений получают:

ШАГ 1

H= 00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

M= 73616820 65676173 73656D20 6C616E69

6769726F 20656874 2065736F 70707553

K₁=73736720 61656965 686D7273 20206F6F

656C2070 67616570 616E6875 73697453

K₂=14477373 0C0C6165 1F01686D 4F002020

4C50656C 04156761 061D616E 1D277369

K₃=CBFF14B8 6D04F30C 26051FFE DFFFB000

35094CAF 72F9FB15 7CF006E2 AB1AE227

K₄=EBACCB00 F7006DFB E5E16905 B0B0DFFF

BA1C3509 FD118DF9 F61B830F F8C554E5

S= FF41797C EEAADAC2 43C9B1DF 2E14681C

EDDC2210 1EE1ADF9 FA67E757 DAFE3AD9

E= F0CEEA4E 368B5A60 C63D96C1 E5B51CD2

A93BEFBD 2634F0AD CBBB69CE ED2D5D9A

ШАГ 2

H= F0CEEA4E 368B5A60 C63D96C1 E5B51CD2

A93BEFBD 2634F0AD CBBB69CE ED2D5D9A

M'=00000000 00000000 00000000 00007365

74796220 3035203D 20687467 6E656C20

K₁=F0C6DDEB CE3D42D3 EA968D1D 4EC19DA9

36E51683 8BB50148 5A6FD031 60B790BA

K₂=16A4C6A9 F9DF3D3B E4FC96EF 5309C1BD

FB68E526 2CDBB534 FE161C83 6F7DD2C8

K₃=C49D846D 1780482C 9086887F C48C9186

9DCB0644 D1E641E5 A02109AF 9D52C7CF

K₄=BDB0C9F0 756E9131 E1F290EA 50E4CBB1

1CAD9536 F4E4B674 99F31E29 70C52AFA

S= 62A07EA5 EF3C3309 2CE1B076 173D48CC

6881EB66 F5C7959F 63FCA1F1 D33C31B8

E=95BEA0BE 88D5AA02 FE3C9D45 436CE821

B8287CB6 2CBC135B 3E399EFE F6576CA9

ШАГ 3

H=95BEA0BE 88D5AA02 FE3C9D45 436CE821

B8287CB6 2CBC135B 3E399EFE F6576CA9

L=0000000000000000 00000000 00000000

STANDART AGENTLIGI
 STANDARTLASHTIRISH,
 STANDARTLASHTIRISH
 VA DAVLAT NAZRATI
 BOSHQARMASI

00000000 00000000 00000000 00000190
 K₁=95FEB83E BE3C2833 A09D7C9E BE45B6FE
 88432CF6 D56CBC57 AAE8136D 02215B39
 K₂=8695FEB8 1BBE3C28 E2A09D7C 48BE45B6
 DA88432C EBD56CBC 7FABE813 F292215B
 K₃=B9799501 141B413C 1EE2A062 0CB74145
 6FDA88BC D0142A6C FA80AA16 15F2FDB1
 K₄=94B97995 7D141B41 C21EE2A0 040CB741
 346FDA88 46D0142A BDFA81AA DC1562FD
 S= D42336E0 2A0A6998 6C65478A 3D08A1B9
 9FDDFF20 4808E863 94FD9D6D F776A7AD
 E= 47E26AFD 3E7278A1 7D473785 06140773
 A3D97E7E A744CB43 08AA4C24 3352C745
 ШАГ 4
 H=47E26AFD 3E7278A1 7D473785 06140773
 A3D97E7E A744CB43 08AA4C24 3352C745
 Σ=73616820 65676173 73656D20 6C61E1CE
 DBE2D48F 509A88B1 40CDE7D6 DED5E173
 K₁=340E7848 83223B67 025AAAAB DDA5F1F2
 5B6AF7ED 1575DE87 19E64326 D2BDF236
 K₂=03DC0ED0 F4CD26BC 8B595F13 F5A4A55E
 A8B063CB ED3D7325 6511662A 7963008D
 K₃=C954EF19 D0779A68 ED73D3FB 7DA5ADDC
 4A9D0277 78EF765B C4731191 7EBB21B1
 K₄=6D12BC47 D9363D19 1E3C696F 28F2DC02
 F2137F37 64E4C18B 69CCFBF8 EF72B7E3
 S=790DD7A1 066544EA 2829563C 3C39D781
 25EF9645 EE2C05DD A5ECAD92 2511A4D1
 E=0852F562 3B89DD57 AEB4781F E54DF14E
 EAFBC135 0613763A 0D770AA6 57BA1A47
 Таким образом, результат хэширования есть
 H=0852F562 3B89DD57 AEB4781F E54DF14E
 EAFBC135 0613763A 0D770AA6 57BA1A47.

Библиография

[1] ГОСТ Р 34.11-94

Информационная технология Криптографическая защита информации. Функция хэширования.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI

УДК:

ОКС 35.040

Группа П 85

Ключевые слова: функция хэширования, хэш-функция, массив, электронная цифровая подпись, однонаправленная функция, криптографический алгоритм

Заместитель директора ГУП
«UNICON.UZ»

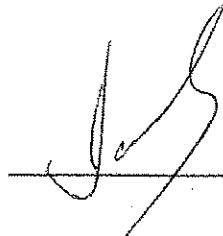
Главный научный сотрудник
ГУП «UNICON.UZ» д.т.н. проф.

Начальник научно-исследовательского
отдела криптографии

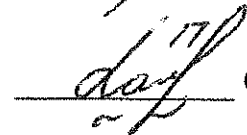
Начальник лаборатории
криптоанализа

Нормоконтроль

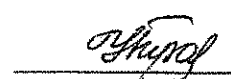
O'z DSt 1106:2009


Х. Хасанов


П. Хасанов


О. Ахмедова


А. Давлатов


Н. Травина

СОГЛАСОВАНО

Служба национальной
безопасности Республики Узбекистан

Письмо от 27 марта 2009 года
№ 39/657

O'ZSTANDART
STANDARTLASHTIRISH
MUVOFIQLASHISH
VA DAVLAT
BOSONBARMASI

1. The first part of the document is a list of the names of the members of the committee who have been appointed to the various sub-committees. The names are listed in alphabetical order of the last name.

Page 1

Page 2