ГОСУДАРСТВЕННЫЙ СТАНДАРТ УЗБЕКИСТАНА

Информационная технология
КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ
Алгоритм шифрования данных

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 28.09.2009 № 05-163

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

4 ВЗАМЕН O‘z DSt 1105:2006

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».

Содержание

1 Область применения 1
2 Нормативная ссылка 1
3 Термины, определения и обозначения 2
3.1 Термины и определения 2
3.2 Обозначения 2
4 Общие положения 3
5 Математические соглашения 4
5.1 Математические определения и предпосылки 4
5.2 Формы представления и соглашения 6
5.3 Параметры и функции алгоритма шифрования 9
6 Основные процессы 11
6.1 Вводные замечания 11
6.2 Процедура зашифрования 11
6.3 Процедура расшифрования 14
6.4 Вопросы реализации 20
Приложение А (справочное) Контрольный пример 22

ГОСУДАРСТВЕННЫЙ СТАНДАРТ УЗБЕКИСТАНА

Ахборот технологияси
АХБОРОТНИНГ КРИПТОГРАФИК МУҲОФАЗАСИ
Маълумотларни шифрлаш алгоритми

Информационная технология
КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ
Алгоритм шифрования данных*

Information technology
CRYPTOGRAPHIC DATA SECURITY
Algorithm of data encryption

Дата введения 2009-10-15
2024-10-15

1 Область применения

Настоящий стандарт «Алгоритм шифрования данных» (АШД) представляет собой криптографический алгоритм, предназначенный для защиты электронных данных. АШД - симметричный блочный шифр, который используется для зашифрования и расшифрования информации. АШД может использовать криптографические ключи длиной ***256*** или ***512*** bit для зашифрования и расшифрования блоков данных длиной ***256*** bit.

Стандарт устанавливает единый алгоритм шифрования информации для систем обработки информации в сетях электронных вычислительных машин (ЭВМ), телекоммуникаций, отдельных вычислительных комплексах и ЭВМ и определяет правила шифрования данных.

Алгоритм шифрования данных предназначен для программной, аппаратной, аппаратно-программной реализации.

Стандарт может быть использован для криптографической защиты данных, хранимых и передаваемых в сетях ЭВМ, телекоммуникаций, в отдельных вычислительных комплексах или в ЭВМ предприятий, организаций и учреждений.

2 Нормативная ссылка

В настоящем стандарте использованы ссылки на следующие стандарты:

---

* **С изменением № 1, 2 утвержденным постановлением агентства Узстандарт от 04.07.2014 № 05-556, от 21.06.2019 № 05-842и**

O‘z DSt 1047:2018 Информационная технология. Термины и определения.

(Новая редакция, Изм. № 1)

O‘z DSt 1109:2013 Информационная технология. Криптографическая защита информации. Термины и определения.

(Измененная редакция, Изм. № 1)

Примечание – При пользовании настоящим стандартом целесообразно проверить действие ссылочного стандартов (классификаторов) 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 **вектор инициализации:** вектор, используемый для определения исходной точки криптографического процесса в рамках криптографического алгоритма.

3.1.2 **сеансовый ключ**: двумерный массив секретных ключей, формируемый на основе ключа шифрования и функционального ключа.

3.1.3 **средства шифрования:** аппаратные, программные и аппаратно-программные средства, реализующие криптографические алгоритмы преобразования информации и предназначенные для защиты информации от несанкционированного доступа при ее обработке, хранении и передаче по каналам связи.

3.1.4 **режим сцепления блоков:** режим шифрования, в котором каждый зашифрованный (расшифрованный) блок зависит от предыдущего зашифрованного (расшифрованного) блока. Для первого блока в качестве предыдущего блока используется вектор инициализации. В случае, если последний блок текста является не полным, он дополняется до необходимой длины.

3.1.5 **режим электронной кодовой книги:** режим шифрования, в котором все блоки открытого текста шифруются независимо друг от друга на одном ключе, в соответствии с алгоритмом шифрования данных.

3.2 Обозначения

В настоящем стандарте использованы следующие обозначения:

M – исходные (открытые) данные (сообщение пользователя);

Издание официальное

C – зашифрованный текст;

m – режим шифрования;

sh – индекс, обозначающий режим преобразования открытого текста в зашифрованный текст;

dsh – индекс, обозначающий режим преобразования зашифрованного текста в открытый текст;

Holat – двумерный массив состояния;

H_{ch}, H_o – левая (верхняя) и правая (нижняя) половина (часть) двумерного массива состояния;

k – ключ шифрования;

k_f – функциональный ключ;

k_{se} – сеансово - этапный ключ;

B_a – линейный массив преобразования элементов на байтовом уровне;

K_{ss} – линейный массив элементов сеансового ключа шифрования;

K_s – двумерный массив сеансового ключа шифрования;

K_{sch}, K_{so} – левая (верхняя) и правая (нижняя) половина (часть) массива сеансового ключа шифрования;

K_e – двумерные массивы этапного ключа;

p, (p+1) – модуль, ***p = 256***;

e – число этапов процедуры зашифрования или расшифрования;

bosqich – порядковый номер этапа процедуры зашифрования или расшифрования;

R – параметр;

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

® – символ операции умножения чисел с коэффициентом ***R*** по модулю ***p***;

$®_2$ – символ операции диаматричного умножения по модулю ***p***;

$\backslash^{-1}$ – символ операции обращения с коэффициентом ***R*** по модулю ***p***;

$\backslash^{d}$ – символ операции возведения в степень ***d*** с параметром ***R*** по модулю ***p***;

$^{-1}$ – символ операции обращения по модулю ***p*** или ***p+1***.;

IV – вектор инициализации.

4 Общие положения

4.1 В симметричных криптосистемах обмен сообщениями происходит в трёх этапах:

- отправитель сообщения передаёт получателю ключ шифрования (или/и функциональный ключ) по защищенному каналу, который никому кроме их самих неизвестен;

- отправитель с помощью ключа шифрования и функционального ключа преобразует исходные данные в зашифрованные данные и отправляет их получателю по каналу связи;

- получатель, получив зашифрованные данные, расшифровывает их с помощью ключа шифрования и функционального ключа. Обе стороны могут воспользоваться этими ключами несколько раз.

4.2 Описываемый АШД основан на использовании преобразований перемешивания столбцов и по байтовой замене, и характеризуется следующими признаками:

- ключи этапа формируются на основе ключа шифрования и функционального ключа, периодически обновляемого через определенное количество сеансов, в зависимости от требуемого уровня защиты информации. Функциональный ключ обновляется на каждом сеансе, когда требуется высокий уровень защиты;

- в перемешивании столбцов участвуют секретные параметры.

Перечисленные признаки обеспечивают повышение криптостойкости алгоритма шифрования данных.

5 Математические соглашения

5.1 Математические определения и предпосылки

5.1.1 Для определения АШД необходимо описать базовые математические объекты, используемые в процессах зашифрования и расшифрования. В данном разделе установлены основные математические определения и требования, накладываемые на параметры алгоритма шифрования данных.

5.1.2 В АШД используется алгебра диаматриц модульной арифметики**,** вычисления в которой осуществляются на том же уровне трудоемкости, что и в алгебре матриц.

5.1.3 Основной операцией алгебры диаматриц, используемой в процедурах зашифрования и расшифрования, является операция обращения диаматрицы в матрицу по модулю ***p***. В этих операциях принимают участие левая или правая половина двумерного массива сеансового ключа, отображаемая квадратной диаматрицей порядка ***4*** **x** ***4*** со специальной структурой, где все диагональные элементы идентичны, и идентичны недиагональные элементы в строке ***1***, а также элементы в начале и конце строки ***2***. Диаматрица специальной структуры порядка ***4*** **x** ***4*** формируется на основе десяти элементов на байтовом уровне. На рисунке 1 приведена диаматрица, сформированная на основе элементов $d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9$.

d_7	d_0	d_1	d_2
d_8	d_7	d_8	d_8
d_9	d_3	d_7	d_9
d_4	d_5	d_6	d_7

Рисунок 1 - Диаматрица специальной структуры

Важным свойством диаматрицы специальной структуры является простота формулы для вычисления диаопределителя диаматрицы, что упрощает процедуру проверки условий обратимости диаматрицы. Обратная диаматрица диаматрицы специальной структуры сохраняет исходную структуру.

5.1.4 Диаопределитель диаматрицы специальной структуры порядка ***4x4*** находится как произведение диагонального элемента на три сомножителя, каждый из которых представляет собой сумму элементов столбца с элементом, расположенным на одной строке с диагональным по соседству справа.

Диаопределитель ***d*** для рассматриваемой диаматрицы находится как:

$$d \equiv d_7 \times (d_7 + d_0 + d_8 + d_3 + d_5) \times (d_7 + d_1 + d_8 + d_9 + d_6) \times (d_7 + d_2 + d_8 + d_9 + d_4) \pmod p. \quad (1)$$

Проверка условий обратимости диаматрицы специальной структуры является основным требованием, предъявляемым к параметрам АШД. Она сводится к сравнению с нулем значений диагонального элемента и упомянутых сомножителей по модулю ***2***. Это позволяет из любого ключа шифрования и функционального ключа формировать обратимую диаматрицу.

5.1.5 В АШД для перемешивания элементов используется операция умножения над диаматрицами порядка ***4x4***:

$$H' \equiv H \circledR_2 K \pmod p,$$

здесь: $\circledR_2$ – символ операции диаматричного умножения по модулю, ***H', H, K*** – диаматрицы порядка ***4x4***, ***H'*** – результирующая диаматрица, ***H, K*** – заданные диаматрицы.

Операция диаматричного умножения $\circledR_2$ выполняется на основе следующих выражений.

Для $s, u \in \{0,1,2,3\}$:

$$h'[u,u] \equiv h[u,u] * \Sigma^3_{i=0} k[i,u] - \Sigma^3_{i=0,\, i\neq u} h[i,i] * k[i,u] \pmod p,$$

$$h'[s,u]_{s\neq u} \equiv h[s,u] * \Sigma^3_{i=0} k[i,u] + k[s,u] * \Sigma^3_{i=0} h[i,u] - \Sigma^3_{i=0;\, i\neq s,u} h[s,i] * k[i,u] \pmod p,$$

здесь: ***h'[u,u]*** – диагональные элементы результирующей диаматрицы, ***h'[s,u]*** - недиагональные элементы результирующей диаматрицы, ***h[s,u], h[i,u], h[s,i], h [i,i], k[i,u], k[s,u,***– элементы ***H, K***.

5.1.6 Также в АШД используются операции алгебры с параметром: умножение, обращение и возведение в степень целых чисел с параметром, алгебры с параметром.

Умножение с параметром ***R*** по модулю ***p*** числа ***X*** на ***Y*** обозначается как $X \circledR Y \pmod p$ и определяется следующим образом:

$$X \circledR Y \pmod p \equiv X + Y (1 + R X) \pmod p. \quad (2)$$

Эта операция является коммутативной и ассоциативной операцией.

Операция обращения с параметром ***R*** переменной по модулю ***p*** числа ***X*** обозначается как $X^{\backslash -1} \pmod p$ и определяется следующим образом:

$$X^{\backslash -1} \pmod p \equiv - X (1 + R X)^{-1} \pmod p, \quad (3)$$

здесь $X^{\backslash -1} \circledR X \equiv 0 \pmod p$, ***0*** – единичный элемент группы с параметром.

Операция возведения в степень ***d*** с параметром ***R*** по модулю ***p*** основания ***X*** выражается следующим образом: $X^{\backslash d}$ ***(mod p)***. Например, если ***d = 37*** возведение в степень с параметром ***R*** по модулю ***p*** основания ***X*** вычисляется следующим образом:

$X^{\backslash 37} \Rightarrow X^{\backslash 32+4+1}$ ***(mod p)*** $\equiv ((((X^{\backslash 2})^{\backslash 2})^{\backslash 2})^{\backslash 2}$ ® $(X^{\backslash 2})^{\backslash 2}$ ® ***X (mod p)***,

здесь: $X^{\backslash 2}$ ***(mod p)*** $\equiv$ ***X (2 + X R) (mod p).***

Используемое в АШД возведение чисел побайтно в степень с параметром выполняется с учетом требования обратимости преобразований.

5.2 Формы представления и соглашения

5.2.1 **Входы и выходы**

5.2.1.1 В АШД как вход, так и выход представляют собой последовательности длиной ***256*** bit, которые иногда будут называться блоками. В АШД используются два ключа - ключ шифрования и функциональный ключ, каждый из которых представляет собой последовательность из ***256*** bit. Исходный функциональный ключ генерируется в двух режимах: 1) как функция от ключа шифрования; 2) независимо от ключа шифрования, т.е. в виде псевдослучайного числа. Обновление функционального ключа осуществляется использованием функции хэширования, где ключом хэширования служит ключ шифрования, входом - функциональный ключ, использованный на предыдущем сеансе. Генерация функционального ключа в зависимости от ключа шифрования длиной ***256*** bit равносильно использованию в АШД ключа шифрования с длиной ***512*** bit. Другие длины входа, выхода, ключа шифрования и функционального ключа указанным стандартом не допускаются.

5.2.1.2 Биты в 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 $0 \le i < 256$ для входных блоков длиной ***256*** bit.

5.2.1.3 В процедурах зашифрования и расшифрования данных алгоритма АШД используются двумерные массивы сеансового и этапных ключей, которые формируются на основе ключа шифрования и функционального ключа.

5.2.1.4 В АШД для обработки входных блоков основной единицей считается байт – последовательность состоящая из восьми bit. Битовые последовательности входных, выходных, сеансовых и этапных ключей обрабатываются как массивы, состоящие из байтов, при этом определенные битовые последовательности формируются посредством разделения на группы, построенные из соседних битов.

5.2.1.5 Байты результирующего массива входного, выходного, сеансово-этапного или этапного ключей, обозначенных буквой ***a,*** применяются с использованием одного из обозначений a_n или ***a*** [***n***], здесь ***n*** принимает значения $0 \le n < 32$.

5.2.1.6 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 $b_7 = 0, b_6 = 1, b_5 = 1, b_4 = 0, b_3 = 0, b_2 = 0, b_1 = 1, b_0 = 1$.

5.2.2 **Массивы байтов**

5.2.2.1 Для входного блока массивы байтов выражаются следующим образом:

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

Байты и расположение битов в байтах определяется ***256*** битовой последовательностью $kir_0, kir_1, kir_2, kir_3, \ldots, kir_{254}, kir_{255}$ следующим образом:

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

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

:

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

Учитывая вышеизложенное на рисунке 2 приведена нумерация битов в байте.

Последовательности битов	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		23
Номер байта	0								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 - Нумерация байтов и битов

5.2.3 **Массив *Holat***

5.2.3.1 Процесс обработки данных для входных блоков в АШД можно интерпретировать как выполнение операций алгоритма над двумерным массивом байтов ***Holat***.

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

В массиве ***Holat***, обозначаемом ***h***, каждый отдельный байт имеет два индекса ***s*** и ***u***, где ***s*** - номер его строки в диапазоне $0 \le s < 4\ (8)$; ***u*** - номер его столбца в диапазоне $0 \le u < 8\ (4)$. Эта индексация позволяет ссылаться на конкретный байт массива ***Holat*** как на ***h [s, u]***. Следующие разъяснения в основном приведены на примере массива размером ***4x8***. Интерпретация относительно массивов размера ***8x4*** аналогична.

5.2.3.2 В самом начале процессов зашифрования и расшифрования, которые будут подробно разбираться позже, входной массив ***kir*** - массив полубайтов $kir_0, kir_1, \ldots kir_{31}$ - копируется в массив ***Holat***, как показано на рисунке 3. Затем в массиве ***Holat*** выполняются необходимые операции за-

шифрования или расшифрования, после чего окончательное значение элементов массива ***Holat*** копируется в выходной массив ***chiq*** - массив полубайтов (байтов) $chiq_o, chiq_1, \ldots chiq_{31}$ (на рисунках 3-7 массивы имеют размер ***4x8***).

Байты массива ***κir***

κir_0	κir_4	κir_8	kir_{12}	kir_{16}	kir_{20}	kir_{24}	kir_{28}
kir_1	κir_5	κir_9	kir_{13}	kir_{17}	kir_{21}	kir_{25}	kir_{29}
κir_2	κir_6	kir_{10}	kir_{14}	kir_{18}	kir_{22}	kir_{26}	kir_{30}
κir_3	κir_7	kir_{11}	kir_{15}	kir_{19}	kir_{23}	kir_{27}	kir_{31}

⇓ Массив ***Holat***

h[0,0]	***h[0,1]***	***h[0,2]***	***h[0,3]***	***h[0,4]***	***h[0,5]***	***h[0,6]***	***h[0,7]***
h[1,0]	***h[1,1]***	***h[1,2]***	***h[1,3]***	***h[1,4]***	***h[1,5]***	***h[1,6]***	***h[1,7]***
h[2,0]	***h[2,1]***	***h[2,2]***	***h[2,3]***	***h[2,4]***	***h[2,5]***	***h[2,6]***	***h[2,7]***
h[3,0]	***h[3,1]***	***h[3,2]***	***h[3,3]***	***h[3,4]***	***h[3,5]***	***h[3,6]***	***h[3,7]***

⇓ Байты массива ***chiq***

$chiq_0$	$chiq_4$	$chiq_8$	$chiq_{12}$	$chiq_{16}$	$chiq_{20}$	$chiq_{24}$	$chiq_{28}$
$chiq_1$	$chiq_5$	$chiq_9$	$chiq_{13}$	$chiq_{17}$	$chiq_{21}$	$chiq_{25}$	$chiq_{29}$
$chiq_2$	$chiq_6$	$chiq_{10}$	$chiq_{14}$	$chiq_{18}$	$chiq_{22}$	$chiq_{26}$	$chiq_{30}$
$chiq_3$	$chiq_7$	$chiq_{11}$	$chiq_{15}$	$chiq_{19}$	$chiq_{23}$	$chiq_{27}$	$chiq_{31}$

Рисунок 3 - Вход и выход массива ***Holat***

Следует отметить, что перед преобразованиями, основанными на использовании операций над матрицами, двумерный массив ***Holat [4,8]*** следует разбивать на левую и правую половины (части): H_{ch} ***[4,4]*** и H_o ***[4,4]***, как это иллюстрировано на рисунке 4.

Массив H_{ch}

$h_{ch}[0,0]$	$h_{ch}[0,1]$	$h_{ch}[0,2]$	$h_{ch}[0,3]$
$h_{ch}[1,0]$	$h_{ch}[1,1]$	$h_{ch}[1,2]$	$h_{ch}[1,3]$
$h_{ch}[2,0]$	$h_{ch}[2,1]$	$h_{ch}[2,2]$	$h_{ch}[2,3]$
$h_{ch}[3,0]$	$h_{ch}[3,1]$	$h_{ch}[3,2]$	$h_{ch}[3,3]$

Массив H_o

$h_o[0,0]$	$h_o[0,1]$	$h_o[0,2]$	$h_o[0,3]$
$h_o[1,0]$	$h_o[1,1]$	$h_o[1,2]$	$h_o[1,3]$
$h_o[2,0]$	$h_o[2,1]$	$h_o[2,2]$	$h_o[2,3]$
$h_o[3,0]$	$h_o[3,1]$	$h_o[3,2]$	$h_o[3,3]$

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

По окончании преобразования эти половины вновь копируются в массив ***Holat [4,8]*** (рисунок 5).

Массив ***Holat***

h [0,0]	***h [0,1]***	***h [0,2]***	***h [0,3]***	***h [0,4]***	***h [0,5]***	***h [0,6]***	***h [0,7]***
h [1,0]	***h [1,1]***	***h [1,2]***	***h [1,3]***	***h [1,4]***	***h [1,5]***	***h [1,6]***	***h [1,7]***
h [2,0]	***h [2,1]***	***h [2,2]***	***h [2,3]***	***h [2,4]***	***h [2,5]***	***h [2,6]***	***h [2,7]***
h [3,0]	***h [3,1]***	***h [3,2]***	***h [3,3]***	***h [3,4]***	***h [3,5]***	***h [3,6]***	***h [3,7]***

Рисунок 5 - Массив ***Holat***

5.2.4 **Массив сеансового ключа K_s**

Процедура формирования массива сеансового ключа K_s начинается с формирования двумерных массивов $K_1[4, 4]$ и $K_2[4, 4]$ из ***20*** левых элементов массива $K_{st}=[32]$, состоящего из байтовых элементов, а также включает в себя операцию обращения диаматрицы в диаматрицу. Процедура формирования сеансового ключа приведена в 6.3.2.

5.2.5 **Массив этапного ключа K_e**

Процедуры зашифрования и расшифрования начинаются с формирования массива сеансового ключа K_s, используемого при побайтовой замене на каждом этапе. Этот массив формируется для каждого этапа на основе сдвига массива линейного этапно-сеансового ключа k_{se}, заданного на битовом уровне.

Массив K_e состоит из четырех (восьми) строк и восьми (четырех) столбцов байтов и к каждому байту массива можно обращаться также, как в массивах ***Holat*** и K_s, т.е. как $K_e[s, u]$.

Результат формирования этапного ключа выражается в виде двумерного массива, как показано на рисунке 6.

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 6 - Массивы этапного ключа K_e

Процедура формирования этапных ключей приведена в 6.3.5.

5.3 Параметры и функции алгоритма шифрования данных

АШД использует следующие параметры и функции:

a) k – ключ шифрования длиной ***256*** или ***512*** bit ;

b) k_f – функциональный ключ длиной ***256*** bit;

c) K_e – двумерный массив этапного ключ порядка ***8x4 (***или ***4x8)***;

d) b – количество входных блоков длиной ***256*** bit;

e) e – количество этапов, $e=8$;

f) ***p, (p +1)*** – модуль, ***p=256***;

g) ***Aralash()*** – является криптографическим преобразованием и выполняется над диаматричными частями при зашифровании и расшифровании; входными данными при зашифровании являются диаматричные части массива ***Holat***, а также массивы K_1 и K_2, выходными данными является массив ***Holat***.

h) ***BaytAlmash()*** – является криптографическим преобразованием и используется для замены на байтовом уровне элементов массива Holat с элементами массива замен; входными данными данного криптопреобразования являются массив ***Holat***, линейный массив замен B_{sA} ***[256]*** или B_{sAD} ***[256]*** на байтовом уровне, а выходными данными является массив ***Holat*** на байтовом уровне;

i) ***Sur()*** – используется при зашифровании и расшифровании для тщательного перемешивания элементов массива ***Holat***; входными данными данного преобразования является массив ***Holat***, при шифровании выходными данными - массив ***Holat*** с циклически сдвинутыми вниз столбацми и с циклически сдвинутыми вправо строками; при расшифровании выходными данными является массив ***Holat*** с циклически сдвинутыми вверх столбацми и с циклически сдвинутыми влево строками;

j) ***ShaklSeansKalitBayt()*** – используется для формирования ключа для каждого сеанса и для выполнения преобразования ***BaytAlmash()*** при зашифровании и расшифровании; в данном преобразовании входными данными являются ключ шифрования ***k*** и функциональный ключ k_f, выходными данными являются массивы B_{sA} ***[256]*** и B_{sAD} ***[256]*** на байтовом уровне;

k) ***ShaklSeansKalit()*** – используется для формирования ключа для каждого сеанса и для выполнения преобразования ***Aralash()*** при зашифровании и расшифровании; входными данными данного преобразования является массив K_{st}***=[32]*** на байтовом уровне; выходными данными являются пара массивов (K_{1t}, K_2) или (K_1, K_{2t}), состоящих из диаматриц специальной структуры;

l) ***ShaklBosqichKalit()*** – используется для формирования из сеансово-этапного ключа этапного ключа и для выполнения преобразования ***Qo’shBosqichKalit()*** при зашифровании и расшифровании; входными данными данного преобразования является линейный массив сеансово-этапного ключа k_{se}, выходными данными является заданный на байтовом уровне двумерный массив K_e***[8,4]***;

m) ***Qo’shBosqichKalit()*** – является простым криптопреобразованием и заключается в выполнении операции исключающего ИЛИ (побитовое сложение по модулю ***2***) при зашифровании и расшифровании массивов ***Holat*** и массива этапного ключа K_e; входными данными данного преобразования являются массивы ***Holat*** и K_e на байтовом уровне, выходными данными является массив ***Holat*** на байтовом уровне.

n) ***Qo’shHolat()*** – является простым криптопреобразованием и выполняется над блоками шифруемых блоков с использованием операции

XOR при зашифровании и расшифровании во всех режимах кроме режима электронной кодовой книги.

6 Основные процессы

6.1 Вводные замечания

6.1.1 В АШД длина входного и выходного блоков, а также длина элементов массива ***Holat*** равны ***256*** bit. Длина ключа шифрования и функционального ключа в АШД также равны ***256*** bit. Для АШД число этапов установлено ***e***=8.

6.1.2 Как в режиме зашифрования, так и в режиме расшифрования алгоритм АШД использует однократное преобразование - формирование массива сеансового ключа и следующих четырех байт- ориентированных и одного бит – ориентированного преобразования на каждом этапе, приведенных в 6.3. К этим преобразованиям относятся:

a) формирование массивов этапных ключей;

b) смешивание данных на основе массива сеансового ключа;

d) циклические сдвиги строк и столбцов массива ***Holat*** на различные значения смещений;

e) побайтовая замена байтов массива ***Holat*** на основе массивов линейного массива;

f) операция сложения по модулю 2 массивов ***Holat*** и массива этапного ключа K_e;

g) циклические сдвиги линейного массива сеансово - этапного ключа на одно и то же значение битов на каждом этапе.

6.2 Процедура зашифрования

6.2.1 Псевдокод процедуры зашифрования

При инициализации шифрующего криптографического модуля (рисунок 7) сначала в криптографический модуль загружаются ключ шифрования ***k*** и функциональный ключ k_f, количество этапов ***e***, а также вектор инициализаци **IV** для режима ***m=ShBil***. Также, при зашифровании в массив Holat криптографического модуля загружается открытый текст, при зашифровании шифртекст. В начале процедуры зашифрования инициализируются криптопреобразования ***ShaklSeansKalitBayt(k,*** k_f***),*** ***ShaklSeansKalit(*** K_{st}***)*** и ***ShaklBosqichKalit(*** k_{se}***)***. На выходах криптопреобразований ***ShaklSeansKalitBayt(k,*** k_f***),*** ***ShaklSeansKalit(*** K_{st}***)*** на байтовом уровне формируются массивы замен и сеансовый ключ, состоящий из диаматричных частей. Эти массивы используются в следующих сеансах до тех пор, пока ***k,*** k_f остаются постоянными. На выходе криптопреобразования ***ShaklBosqichKalit(*** k_{se}***)*** формируется начальный ключ и набор этапных ключей, сформированный для каждого этапа.

Псевдокод для режима электронной кодовой книги **(Elektron kod kitobi**) ***m=Ekk*** и режима сцепления блоков (**ShifrBloklarni ilaktirish**) ***m=ShBil*** приведен на рисунке 7.

6.2.2 Простые криптопреобразования ***Aralash(Holat,K_s), BaytAlmash(Holat,B_a), Qo’shBosqichKalit(Holat, K_e), Sur(Holat)*** и преобразования ***ShaklSeansKalitBayt(k,k_f), ShaklSeansKalit(K_{st}), ShaklBosqichKalit(k_{se})*** и ***Qo’shHolat(Holatm, Holat)*** приведенеы в следующем подразделе.

Обновление функционального ключа в аппаратно-программном модуле целесообразно выполнять в совокупности с преобразованиями ***ShaklSeansKalitBayt(k,k_f), ShaklSeansKalit (K_{st}), ShaklBosqichKalit(k_{se})***. В этом случае в псевдокод шифрования необходимо ввести результаты ***ShaklSeansKalitBayt(k,k_f), ShaklSeansKalit (K_{st}), ShaklBosqichKalit(k_{se})***.

В приложении А в качестве примера приведен псевдокод процедуры зашифрования ***(sh)*** и расшифрования ***(dsh)*** для базовых режимов **Elektron kod kitobi** и **ShifrBloklarni ilaktirish**, здесь приведены значения массива ***Holat*** после применения в процессе каждого этапа вышеупомянутых преобразований.

```
Shifr (int blok_soni, byte IV[32], byte kirish [block_soni][32],
byte chiqish [blok_soni][32], byte k[32], byte kf[32], byte e)
begin
     byte ke [8,4], Ks [8,4], Ke [8,4]
     Holat [8,4],  Holatn[8,4]
   if(m=Sh)
     ShaklSeansKalitBayt (k, kf)
     ShaklSeansKalit(Kst)
     ShaklBosqichKalit(kse)
     for  blok =1 step 1 to blok_soni
          Holat=kirish[blok]
          if(m=ShBil)
             if(blok=1)
                Holatn=IV
             else
                Holatn=chiqish[blok-1]
             end if
             Qo’shHolat(Holat, Holatn)
          end if
          for  bosqich =1 step 1 to e
               Qo’shBosqichKalit(Holat, Ke)
               Aralash(Holat, Ks)
               Sur(Holat)
               BaytAlmash(Holat, Ba)
          end for
          Qo’shBosqichKalit(Holat, Ke)
          Aralash(Holat, Ks)
          chiqish[blok]= Holat
     end for
   else
     ShaklSeansKalitBayt (k, kf)
     ShaklSeansKalit(Kst)
     ShaklBosqichKalit(kse)
     for  blok =1 step 1 to blok_soni
          Holat=kirish[blok]
          Aralash(Holat, Ks)
          Qo’shBosqichKalit(Holat, Ke)
          for  bosqich =1 step 1 to e
               BaytAlmash(Holat, Ba)
               Sur(Holat)
               Aralash(Holat, Ks)
               Qo’shBosqichKalit(Holat, Ke)
          end for
          if(m=ShBil)
             if(blok=1)
                Holatn=IV
             else
               Holatn = kirish[blok-1]
             end if
             Qo’shHolat(Holat, Holatn)
          end if
          chiqish[blok]= Holat
     end for
   end if
end
```

Рмсунок 7 – Псевдокод процедуры шифрования

6.3 Криптографические преобразования

6.3.1 Преобразование ***ShaklSeansKalitBayt(k,k_f)***

Преобразование ***ShaklSeansKalitBayt(k,k_f)*** заключается в выполнении следующих операций.

6.3.1.1 Вычислить $k_{se}= k+k'*(1+k_{f*} k)$ и оставить ***672*** bit слева, здесь ***k'-*** это ***192*** bit k_f справа.

Примечание - При генерации ключа шифрования, набора функциональных ключей, обновляемых с использованием ключа шифрования, и при формировании на их основе ключа k_{se} они обязательно должны быть проверены на основе критериев случайности и с левой части результирующего k_{se}, начиная с первого левого бита выделяются *672 bit*.

6.3.1.2 Выделить с правой части k_{se} ***256+64*** bit, из левой ***256*** битной части сформировать линейный массив $K_{st}=[0,1,2,3,\dots\ 31]$, состоящий из байтовых элементов**,** из остальной ***64*** битной части сформировать линейный массив ***B= [0,1,2,3,4,5,6,7]***, состоящий из элементов на байтовом уровне.

6.3.1.3 Из элементов линейного массива ***B*** сформировать пару массивов $B_1= [0,1,2,3]$ и $B_2= [4,5,6,7]$ и на основе нижеперечисленных правил сформировать тройку параметров (d_1, R_1, L_1) и (d_2, R_2, L_2):

1) для $j=0$, если $b[j] <3$, принять $d_1=3$, в противном случае принять $d_1= b[0]$; для $j=4$, если $b[j] <3$, принять $d_2=3$, в противном случае принять $d_2= b[4]$.

2) для $j=1$, если $b[j]=0$, принять $R_1=1$, в противном случае принять $R_1= b[1]$; для $j=5$, если $b[j]=0$, принять $R_2=1$, в противном случае $R_2= b[5]$.

3) для $j=2$, если $b[j]=0$, принять $L_1=1$, в противном случае принять $L_1= b[2]$; для $j=6$, если $b[j]=0$, принять $L_2=1$, в противном случае принять $L_2= b[6]$.

4) для $d_s \pmod 2=0$, если $d_s \pmod 4=0$, принять $d_s= d_s -1$, в противном случае принять $d_s = d_s +1$, здесь $s\in\{1,2\}$.

5) для $d_s \pmod 2=1$, если $d_s -1 \pmod 4=0$, тогда принять $d_s = d_s-2$.

6) для $j=3$, если $b[j] =0$, принять $b[j] =1$; для $j=7$, если $b[j] =0$, принять $b[j] =1$.

6.3.1.4 Для выполнения преобразований на байтовом уровне и получения шифртекста сформировать пару массивов $(B_{1A}\ [256],\ B_{2A}\ [256])$. Смысл формирования этих массивов разъясняется в нижеследующих указаниях:

- для $s\in\{1,2\}$ возвести в степень d_s с параметром R_s по модулю ***257*** значение $((i+L_s) \bmod 256)+1$, соответствующее каждому адресу $i\in\{0,1,2,\dots 255\}$, результат привести по модулю ***256*** и в каждом шаге текущее значение сравнить со значением предыдущего шага и i. Если значения равны или, если текущее значение близко $(|b_{sA}\ [i-1]-b_{sA}\ [i]\ | \geq 8)$ к предыдущему значению, то это текущее значение заменить на ($i-b[3]$) при $s=1$ и на ($i-b[7]$) при $s=2$.

Примечание - В псевдокоде в нечетных этапах используется массив $\mathbf{B_{1A}\ [256]}$, а в четных этапах массив $\mathbf{B_{2A}\ [256]}$.

Алгоритм вычислений включает в себя следующие операции:

1) вычислить $b_{sA}[i] \equiv (((i+L) \bmod 256)+1)^{|ds} \pmod{257} \pmod{256}$ для $i=0 \div 255$.

Начиная с $i=1$ на каждом шаге проверить условие $i-b_{sA}[i] \neq 0$ и $|b_{sA}[i-1]-b_{sA}[i]| \geq 8$ и если оба условия выполняются, то принять значение, в противном случае при $s=1$ поменять местами $b_{sA}[i]$ и элемент, находящийся по адресу ($i-b[3]$) $\pmod{256}$ или при $s=2$ поменять местами $b_{sA}[i]$ и элемент, находящийся по адресу $b[7]= b[7]-5 \pmod{256}$ и принять $b[3]= b[3]-5 \pmod{256}$ для $s=1$ или $b[7]= b[7]-5 \pmod{256}$ для $s=2$.

2) из элементов $b_{sA}[i,\ i=0,1,2,\ldots 255]$, чтобы получить шифртекст на байтовом уровне, т.е. для использования в режиме ***sh*** сформировать пару линейных массивов $(B_{1A}[256], B_{2A}[256])$.

6.3.1.5 Для расшифрования на байтовом уровне, т.е. для использования в режиме ***dsh*** сформировать пару линейных массивов $(B_{1AD}[256], B_{2AD}[256])$.

Для этого, чтобы получить шифртекст (***dsh***) на байтовом уровне, достаточно заменить каждый элемент $b_{sA}[i]$ на значение в линейном массиве $B_{sA}[256]$ с индексом, равным его значению, здесь $s \in \{1,2\}$ и расположить элементы сформированного массива в порядке возрастания адреса.

Примечание - В псевдокоде в нечетных этапах используется массив $B_{2AD}[256]$, в четных этапах массив $B_{1AD}[256]$.

6.3.2 Преобразование ***ShaklSeansKalit*** (K_{st})

Преобразование ***ShaklSeansKalit*** (K_{st}) заключается в выполнении следующих операций.

6.3.2.1 Сформировать линейный массив $K_{ss}=[0,1,2,3,\ldots 19]$, состоящий из ***20*** байтовых элементов слева линейного массива $K_{st}=[0,1,2,3,\ldots 31]$, состоящего из байтовых элементов.

Если $k_{ss}[i] = 0$ для $i = 0 – 19$, заменить $k_{ss}[i]$ на $k_{ss}[i]–1 \pmod p$.

Если $k_{ss}[6] \pmod 2 = 0$, заменить $k_{ss}[6]$ на $k_{ss}[6]–1 \pmod p$.

Если $k_{ss}[16] \pmod 2 = 0$, заменить $k_{ss}[16]$ на $k_{ss}[16]–1 \pmod p$.

Если $k_{ss}[6]+k_{ss}[0]+k_{ss}[8]+k_{ss}[3]+k_{ss}[5] \pmod 2 = 0$, заменить $k_{ss}[8]$ на $k_{ss}[8]–1 \pmod p$.

Если $k_{ss}[16]+k_{ss}[10]+k_{ss}[18]+k_{ss}[13]+k_{ss}[15] \pmod 2 = 0$, заменить $k_{ss}[18]$ на $k_{ss}[18] –1 \pmod p$.

Если $k_{ss}[6]+k_{ss}[1]+k_{ss}[3]+k_{ss}[9]+k_{ss}[4] \pmod 2 = 0$, заменить $k_{ss}[9]$ на $k_{ss}[9]–1 \pmod p$.

Если $k_{ss}[16]+k_{ss}[11]+k_{ss}[13]+k_{ss}[19]+k_{ss}[14] \pmod 2 = 0$, заменить $k_{ss}[19]$ на $k_{ss}[19] –1 \pmod p$.

Если $k_{ss}[6]+k_{ss}[2]+k_{ss}[3]+k_{ss}[9]+k_{ss}[7] \pmod 2 = 0$, заменить $k_{ss}[7]$ на $k_{ss}[7]–1 \pmod p$.

Если $k_{ss}[16] + k_{ss}[12] + k_{ss}[13] + k_{ss}[19] + k_{ss}[17] \pmod 2 = 0$, заменить $k_{ss}[17]$ на $k_{ss}[17] –1 \pmod p$.

6.3.2.2 Из элементов линейного массива K_{ss} в следующем порядке сформировать двумерные массивы $K_1[4, 4]$ и $K_2[4, 4]$:

1) разделить линейный массив $K_{ss}=[0,1,2,3,\ldots19]$ на два линейных массива $K_{ss1}=[0,1,2,3,\ldots9]$ и $K_{ss2}=[10,11,12,13\ldots,19]$ и каждыый из них однозначно отобразить в упорядоченный набор $\{k_{s1}[0,1], k_{s1}[0,2], k_{s1}[0,3], k_{s1}[1,0], k_{s1}[2,0], k_{s1}[2,1], k_{s1}[2,2], k_{s1}[3,0], k_{s1}[3,1], k_{s1}[3,2]\}$ и $\{k_{s2}[0,1], k_{s2}[0,2], k_{s2}[0,3], k_{s2}[1,0], k_{s2}[2,0], k_{s2}[2,1], k_{s2}[2,2], k_{s2}[3,0], k_{s2}[3,1], k_{s2}[3,2]\}$ и соответственно из каждого из них сформировать элементы $k_1[i,j], k_2[i,j]$ массивов $K_1[4, 4]$ и $K_2[4, 4]$;

2) сформировать остальные элементы массива $K_s[4, 4], s\in\{1,2\}$ на основе следующего правила:

- для $j\in\{0,1,2,3\}$, если $i=j$, то соответствующие элементы равны по значению $k_s[2,2]$;

- для $i=1, j=0,2,3$ соответствующие элементы равны по значению $k_s[1,0]$;

- для $i=2, j=0,3$ соответствующие элементы равны по значению $k_s[2,0]$.

В результате для использования в режиме шифрования в качестве $K_s[8, 4]$ формируется пара диаматриц специальной структуры $K_1[4, 4]$ и $K_2[4, 4]$.

В результате для использования в режиме ***sh*** в качестве $K_s[8, 4]$ формируется две диаматрицы специальной структуры $K_1[4, 4]$ и $K_2[4, 4]$.

$k_{ss}[6]$	$k_{ss}[0]$	$k_{ss}[1]$	$k_{ss}[2]$
$k_{ss}[3]$	$k_{ss}[6]$	$k_{ss}[3]$	$k_{ss}[3]$
$k_{ss}[4]$	$k_{ss}[5]$	$k_{ss}[6]$	$k_{ss}[4]$
$k_{ss}[7]$	$k_{ss}[8]$	$k_{ss}[9]$	$k_{ss}[6]$

$k_{ss}[16]$	$k_{ss}[10]$	$k_{ss}[11]$	$k_{ss}[12]$
$k_{ss}[13]$	$k_{ss}[16]$	$k_{ss}[13]$	$k_{ss}[13]$
$k_{ss}[14]$	$k_{ss}[15]$	$k_{ss}[16]$	$k_{ss}[14]$
$k_{ss}[17]$	$k_{ss}[18]$	$k_{ss}[19]$	$k_{ss}[16]$

↓

Массив K_1

$k_1[0,0]$	$k_1[0,1]$	$k_1[0,2]$	$k_1[0,3]$
$k_1[1,0]$	$k_1[1,1]$	$k_1[1,2]$	$k_1[1,3]$
$k_1[2,0]$	$k_1[2,1]$	$k_1[2,2]$	$k_1[2,3]$
$k_1[3,0]$	$k_1[3,1]$	$k_1[3,2]$	$k_1[3,3]$

Массив K_2

$k_2[0,0]$	$k_2[0,1]$	$k_2[0,2]$	$k_2[0,3]$
$k_2[1,0]$	$k_2[1,1]$	$k_2[1,2]$	$k_2[1,3]$
$k_2[2,0]$	$k_2[2,1]$	$k_2[2,2]$	$k_2[2,3]$
$k_2[3,0]$	$k_2[3,1]$	$k_2[3,2]$	$k_2[3,3]$

6.3.2.3 Для матрицы специальной структуры $K_1[4, 4]$ вычислить обратную матрицу специальной структуры $K_{1t}[4, 4]$ для использования в режиме ***sh***.

Для матрицы специальной структуры $K_2[4, 4]$ вычислить обратную матрицу специальной структуры $K_{2t}[4, 4]$ для использования в режиме ***dsh***.

Получение обратной матрицы для диаматрицы специальной структуры $K_{1i}[4, 4]$ (здесь $i=\{1,2\}$), у которой диаопределитель равен нулю, заключается в вычислении обратной матрицы для матрицы, полученной в результате выполнения диапреобразования над ним и в результате выполнения диапреобразования над полученной обратной матрицей.

В результате обращения диаматриц специальной структуры $K_1[4, 4]$ и $K_2[4, 4]$ образуются диаматрицы специальной структуры $K_{1t}[4, 4]$ и $K_{2t}[4, 4]$, имеющие следующий вид (K_{1t} и K_{2t}).

Массив K_{1t}

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

Массив K_{2t}

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

В режиме ***shy f*** на вход подается пара, сосотоящая из (K_{1t}, K_2), в режиме ***dsh*** на вход подается пара сосотоящая из (K_1, K_{2t}).

6.3.3 Преобразование ***Aralash (Holat, K_s)***

Преобразование ***Aralash (Holat, K_s)*** заключается в выполнении следующих операций.

Если $m=sh$, то принять $K_1=K_{1t}$, $K_2=K_2$, вычислить $H_1 ®_2 K_1 \pmod p$, $H_2 ®_2 K_2 \pmod p$, результат записать в массивы H_1, H_2 и копировать в массив ***Holat***, в противном случае, т.е. если $m=dsh$, то принять $K_1=K_1$, $K_2=K_{2t}$, вычислить $H_1 ®_2 K_1 \pmod p$, $H_2 ®_2 K_2 \pmod p$, результат записать в массивы H_1, H_2 и копировать в массив ***Holat***.

Операция диаматричного умножения $®_2$ выполняется на основе нижеследующих выражений, здесь индекс, используемый в выражениях, принимает значения $s\in\{1,2\}$.

Выражения для $i=j\in\{0,1,2,3\}$:

$$h'_s[0,0]=h_s[0,0](k_s[0,0]+k_s[1,0]+k_s[2,0]+k_s[3,0])-h_s[1,1]k_s[1,0]-$$
$$-h_s[2,2]k_s[2,0]-h_s[3,3]k_s[3,0] \pmod p,$$

$$h'_s[1,1)= h_s[1,1] (k_s[0,1]+k_s[1,1]+k_s[2,1]+k_s[3,1])-h_s[0,0]k_s[0,1]-$$
$$-h_s[2,2]k_s[2,1] -h_s[3,3]k_s[3,1] \pmod p,$$

$$h'_s[2,2)= h_s[2,2](k_s[0,2]+k_s[1,2]+k_s[2,2]+k_s[3,2])-h_s[0,0]\ k_s[0,2]-$$
$$-h_s[1,1]*k_s[1,2]-h_s[3,3]k_s[3,2] \pmod p,$$

$$h'_s[3,3)= h_s[3,3](k_s[0,3]+k_s[1,3]+k_s[2,3]+k_s[3,3])-h_s[0,0]k_s[0,3]-$$
$$-h_s[1,1]k_s[1,3]-h_s[2,2]k_s[2,3] \pmod p.$$

Выражения для $i \neq j \in \{0,1,2,3\}$:

$$h'_s[0,1]=h_s[0,1](k_s[0,1]+k_s[1,1]+k_s[2,1]+k_s[3,1])+(h_s[0,0]+h_s[1,0]+ +h_s[2,0]+h_s[3,0])k_s[0,1]-h_s[0,2]k_s[2,1]-h_s[0,3]\,k_s[3,1] \pmod p,$$

$$h'_s[0,2]=h_s[0,2](k_s[0,2]+k_s[1,2]+k_s[2,2]+k_s[3,2])+(h_s[0,0]+h_s[1,0]+ +h_s[2,0]+h_s[3,0])k_s[0,2] - h_s[0,1]\,k_s[1,2] - h_s[0,3]k_s[3,2] \pmod p,$$

$$h'_s[0,3]=h_s[0,3](k_s[0,3]+k_s[1,3]+k_s[2,3]+k_s[3,3])+(h_s[0,0]+h_s[1,0]+ +h_s[2,0]+h_s[3,0])k_s[0,3] - h_s[0,1]\,k_s[1,3] - h_s[0,2]\,k_s[2,3] \pmod p,$$

$$h'_s[1,0]=h_s[1,0](k_s[0,0]+k_s[1,0]+k_s[2,0]+k_s[3,0])+(h_s[0,1]+h_s[1,1]+ +h_s[2,1]+h_s[3,1])k_s[1,0] - h_s[1,2]\,k_s[2,0] - h_s[1,3]\,k_s[3,0] \pmod p,$$

$$h'_s[1,2]=h_s[1,2](k_s[0,2]+k_s[1,2]+k_s[2,2]+k_s[3,2])+(h_s[0,1]+h_s[1,1]+ +h_s[2,1]+h_s[3,1])k_s[1,2] - h_s[1,0]\,k_s[0,2] - h_s[1,3]\,k_s[3,2] \pmod p,$$

$$h'_s[1,3]=h_s[1,3](k_s[0,3]+k_s[1,3]+k_s[2,3]+k_s[3,3])+(h_s[0,1]+h_s[1,1]+ +h_s[2,\ 1]+\ h_s[3,\ 1])k_s[1,3] - h_s[1,0]\,k_s[0,3] - h_s[1,2]\,k_s[2,3] \pmod p,$$

$$h'_s[2,0]=h_s[2,0](k_s[0,0]+k_s[1,0]+k_s[2,0]+k_s[3,0])+(h_s[0,2]+h_s[1,2]+ +h_s[2,2]+h_s[3,\ 2])k_s[2,0] - h_s[2,1]\,k_s[1,0] - h_s[2,3]\,k_s[3,0] \pmod p),$$

$$h'_s[2,1]=h_s[2,1](k_s[0,1]+k_s[1,1]+k_s[2,1]+k_s[3,1])+(h_s[0,2]+h_s[1,2]+ +h_s[2,\ 2]+h_s[3,\ 2])\ \ k_s[2,1] - h_s[2,0]\ \ k_s[0,1] - h_s[2,3]\ \ k_s[3,1] \pmod p,$$

$$h'_s[2,3]=h_s[2,3](k_s[0,3]+k_s[1,3]+k_s[2,3]+k_s[3,3])+(h_s[0,2]+h_s[1,2]+ +h_s[2,2]+h_s[3,2])\ k_s[2,3] - h_s[2,0]\ \ k_s[0,3] - h_s[2,1]\ \ k_s[1,3] \pmod p,$$

$$h'_s[3,0]=h_s[3,0](k_s[0,0]+k_s[1,0]+k_s[2,0]+k_s[3,0])+(h_s[0,3]+h_s[1,3]+ +h_s[2,3]+h_s[3,3])\ \ k_s[3,0] - h_s[3,1]\ \ k_s[1,0] - h_s[3,2]\ \ k_s[2,0] \pmod p,$$

$$h'_s[3,1]=h_s[3,1](k_s[0,1]+k_s[1,1]+k_s[2,1]+k_s[3,1])+(h_s[0,3]+h_s[1,3]+ +h_s[2,3]+h_s[3,3])\ k_s[3,1] - h_s[3,0]\ \ k_s[0,1] - h_s[3,2]\ \ k_s[2,1] \pmod p,$$

$$h'_s[3,2]=h_s[3,2](k_s[0,2]+k_s[1,2]+k_s[2,2]+k_s[3,2])+(h_s[0,3]+h_s[1,3]+ +h_s[2,3]+h_s[3,\ 3])\ k_s[3,2] - h_s[3,0]\ \ k_s[0,2] - h_s[3,1]\ \ k_s[1,2] \pmod p.$$

Данное преобразование эффективнее по сравнению с матричным преобразованием. Здесь, при изменении одного элемента в исходном массиве ***Holat***, в зависимости от адреса изменившегося элемента, изменяется ***6*** или ***7*** элементов.

6.3.4 Преобразование ***BaytAlmash(Holat, B_a)***

Преобразование ***BaytAlmash(Holat,B_a)*** заключается в выполнении следующих операций.

1) Назвать массив ***Holat[8,4],*** заданный на байтовом уровне, как массив ***Holatb[8, 4]*** на байтовом уровне.

2) Если ***m=sh***, то принять ***B_a[256] = B_{sA}[256]***; заменить каждый элемент мсассива ***Holatb[8, 4]*** элементом массива ***B_a***, находящегося по адресу, равному значению элемента массива ***Holatb[8, 4]*** и присвоить результирующий массив ***Holatb[8, 4]*** массиву ***Holat[8, 4]***, заданного на байтовом уровне, в противном случае, если ***m=dsh***, принять ***B_a[256] = B_{sAD} [256]***, заменить каждый элемент массива ***Holatb[8, 4]*** элементом массива ***B_a***, находящимся по адресу равному значению элемента массива ***Holatb[8, 4]*** и присвоить результирующий массив ***Holatb[8, 4]*** массиву ***Holat[8, 4]***, заданному на байтовом уровне, здесь ***$s \in \{1,2\}$***.

6.3.5 Преобразование ***ShaklBosqichKalit (k_{se})***

Преобразование ***ShaklBosqichKalit (k_{se})*** (формирование линейного сеансово-этапного ключа) происходит следующим образом:

1) если ***bosqich=1*** и ***m=sh***, то оставить неизмененным массив k_{se} линейного сеансово-этапного ключа, если ***bosqich=1*** и ***m=dsh***, то сдвинуть массив k_{se} на ***672-(e x 83) mod 672*** bit влево.

Отделить левую ***256*** битовую часть массива линейного сеансово-этапного ключа и сформировать из неё на байтовом уровне массив $K_e[8,4]$. Это преобразование выполняется на всех этапах до начала процедуры шифрования.

2) если ***bosqich>1*** и ***m=sh***, то массив k_{se} циклически сдвинуть на ***83*** bit влево; если ***bosqich>1*** и ***m=dsh***, то массив k_{se} циклически сдвинуть на ***83*** bit вправо.

Отделить левую ***256*** битовую часть массива линейного сеансово-этапного ключа и сформировать из неё на байтовом уровне массив $K_e[8,4]$.

Это преобразование выполняется на всех этапах до начала процедуры расшифрования.

6.3.6 Преобразование ***Qo’shBosqichKalit (Holat, K_e)***

Преобразование ***Qo’shBosqichKalit (Holat, K_e)*** заключается в выполнении операции исключающего ИЛИ (побитовое сложение по модулю 2) над одноименными элементами на байтовом уровне массивов ***Holat*** и $K_e[8,4]$.

Для $0 \le c < 8$:

$[h'[c,0], h'[c,1], h'[c,2], h'[c,3]]=[h[c,0], h[c,1], h[c,2], h[c,3]] \oplus [k_e[c,0], k_e[c,1], k_e[c,2], k_e[c,3]]$.

Результат скопировать в массив ***Holat***.

6.3.7 Преобразование ***Sur (Holat)***

Если ***m= sh***, то сначала циклически сдвинуть ***j***-столбец массива ***Holat*** на ***(j+1) (mod 8)*** байтов вниз, затем сдвинуть ***i***-строку результирующего массива на ***(i+1) (mod 4)*** байтов вправо, в противном случае, т.е. если ***m=dsh***, то сначала циклически сдвинуть ***i***-строку массива ***Holat*** на ***(i+1) (mod 4)*** байтов влево, затем сдвинуть ***j***-столбец результирующего массива на ***(j+1) (mod 8)*** байтов вверх. Здесь, $0 \le i < 4$, $0 \le j < 8$.

6.3.8 Преобразование ***Qo’shHolat(Holatn, Holat)***

В преобразовании ***Qo’shHolat(Holatn, Holat)*** к каждому байту массива ***Holatn*** прибавляется побитово с помощью операции сложения ***XOR*** (сложение по модулю ***2***) одноименный байт массива ***Holat***. Массив ***Holatn*** состоит восьми слов. Когда эти слова в диапазоне $0 \le s < 8$, то элементы массива ***Holat***, находящиеся, в столбцах складываются по отдельности следующим образом:

[h'[s,0], h'[s,1], h'[s,2], h'[s,3]]=[h[s,0], h[s,1], h[s,2], h[s,3]] $\oplus$ ***[hn[s,0], hn[s,1], hn[s,2], hn[s,3]],***

здесь: ***hn –*** элементы массива ***Holatn***, ***h'*** – элементы результирующего массива.

Результат преобразования копируется в массив ***Holat***.

6.4 Вопросы реализации

АШД реализуется с использованием ключей длиной в ***256*** и ***512*** bit.

В первом случае в криптографический модуль, который используется для шифрования, вводится ***256*** битный ключ. Этот ключ полностью используется как ключ шифрования ***k***, а функциональный ключ k_f начального сеанса вычисляется как значение хэш-функции.

Во втором случае в криптографический модуль, который используется для шифрования, вводится ***512*** битный ключ. Первая 256 битная часть этого ключа используется как ключ шифрования ***k***, вторая часть используется как функциональный ключ k_f первого сеанса.

В третьем случае в криптографический модуль, который используется для шифрования, не вводится никакой ключ. В качестве ключа шифрования используется ключ шифрования ***k***, использованный на предыдущем этапе, а в качестве функционального ключа k_f используется хэшированное значение с использованием ключа шифрования функционального ключа k_{f-1}, использованного на предыдущем этапе.

В первом и втором случае, рассмотренных выше, обновленный функциональный ключ k_f очередного сеанса вычисляется как хэш-функция функционального ключа k_{f-1} предыдущего сеанса. В качестве ключа хэширования по правилу используется ключ шифрования, а программа вычисления функции хэширования включается в программное (аппаратное) обеспечение АШД. Период обновления функционального ключа определяется протоколом, использующим АШД в зависимости от режима использования АШД и от уровня конфиденциальности шифруемой информации.

Стандарт АШД может применяться в любых режимах блочных алгоритмов шифрования, установленных в международных стандартах. В данном стандарте описаны два базовых режима:

- электронная кодовая книга (**Elektron kod kitobi**);
- сцебление блоков (**ShifrBloklarni ilaktirish**).

В режиме «электронная кодовая книга» все блоки открытого текста шифруются независимо друг от друга посредством замены каждого блока открытого текста блоком шифртекста (каждого блока шифртекста блоком открытого текста).

Режим «сцепление блоков» использует механизм **обратной связи** и имеет специальный вход для вектора инициализации **IV**, обновляемого на каждом сеансе: в начале псевдокода зашифрования над текущим блоком открытого текста и над предыдущим блоком шифртекста выполняется

преобразование ***Qo’shHolat(Holatn, Holat)***; в конце псевдокода расшифрования над текущим блоком зашифрованного текста и предыдущим блоком расшифрованного текста выполняется преобразование ***Qo’shHolat(Holatn, Holat)***.

Приложение А
(справочное)
Контрольный пример

В данном приложении приведены процедуры зашифрования и расшифрования информации, состоящей из одного блока, представленного в виде чисел. Здесь числа приведены в шестнадцатеричной системе счисления.

Входной блок (Открытый текст):

30	31	32	33
34	35	36	37
38	39	41	42
43	44	45	46
30	31	32	33
34	35	36	37
38	39	41	42
43	44	45	46

Ключ шифрования K:
37 B6 0B BA 0A B1 60 CF DC 18 F5 0C DE E8 E0 45 30 B3 F8 AF 14 32 FE 51 1F BB 20 29 11 2F 21 43

Функциональный ключ Kf:
47 E7 69 46 69 C5 46 B6 FE 16 3A 89 B0 D8 96 D6 23 8B 23 15 32 C4 04 34 9C B0 C7 AA 81 3D F9 6D

Вектор инициализации IV:
26 54 BB 5F A3 75 D8 98 54 EA 48 9F 9A A8 84 16 FD 4D EB BD 9B 3B 40 33 48 29 F9 EE 52 34 C3 7A

Массив замены байтов Bsa1
62 E9 A9 E5 6C 9B 02 4D B3 EB AB E3 48 93 B9 29 B1 00 AD 33 F5 A8 FD 58 04 8D C4 26 5D CE 8A 35 F3 54 69 61 71 66 3C 89 51 2B D1 22 A5 6E E2 25 30 68 E1 21 D5 1F 7C 1D 79 9F 1A 90 7D CC 86 15 81 9D EA 84 5C 16 BA 1B E7 BF 45 FC C2 52 BC E0 6D E8 47 94 DE B2 9A 4E 03 8E 5E B8 56 D8 D6 F6 23 A2 31 A4 28 A6 D7 13 42 D9 C3 46 1C 57 40 B0 32 67 4C 10 50 77 DF FA 12 06 C7 87 AA FF 0B 64 75 EF C9 3A 6B D0 A1 DD 4A 0A CB 8B 4B FB ED C6 2A 5B CD 59 C1 88 5F 97 EC 0E CF 8F 83 9C B7 1E BB 37 49 2E 63 E6 0C 3E BE 4F D3 BD 53 82 3B F0 E4 F2 A3 F4 72 A0 C0 98 3F 95 C8 07 92 F8 09 17 01 9E AE 7E B6 DA FE D4 8C DB 5A AF 7A DC 7F 39 11 AC F9 14 80 0F 96 18 05 7B F7 43 78 19 76 20 74 B5 2D 65 70 B4 27 F1 73 C5 6A 44 55 A7 2F 24 34 2C 41 0D 60 36 85 38 6F D2 08 91 3D 99 EE CA

********Процесс шифрования********

Массив замены байтов Bsa2
E9 47 0C 68 0A 38 08 15 4E DD 88 BD 4C 9D E7 7D FF CB FD 17 27 5B 26 57 1F 77 11 BE A7 63 14 65 EF 67 00 3E EB 36 46 E8 3A 76 E5 BF 72 E2 8F 7F E6 F6 4F 79 2F 20 0F 49 7E E0 D0 C0 2B 83 C5 80 CF 60 04 29 24 DA 92 D3 8E E1 84 5C 8C B4 C4 81 71 61 51 BC 23 9C 9B 01 9E B6 A0 93 B3 F3 A4 B0 A6 B2 52 12 AB C2 42 02 62 AF A5 B1 A3 91 AA B5 9F B7 53 B9 50 31 99 03 F4 E4 D4 A2 7B C3 C6 D1 74 C7 1E C9 34 F2 89 BA F5 86 D5 90 D2 82 95 85 75 D7 55 45 A1 25 DC 05 5F 6F D6 C1 73 E3 96 70 DF 6E 40 0E C8 6A 16 ED 35 66 F0 FB 30 AE 97 87 DE F7 33 F9 1A 2D 3D 58 43 56 D8 B8 A8 98 CD 78 06 4D AD 1C 4A 0B 48 0D 6C D9 10 CE AC 41 6D 07 69 FC 18 3B 54 39 4B 1D 6B EE CA 21 32 BB F8 7A 2E F1 2C 8D 2A CC 28 5A 94 DB 8A 9A 22 FE 8B 5E EA 37 09 1B FA 19 3C 59 EC 3F 64 13 A9 5D 44 7C

Shakl seans Kalit Kse
F8 7E 98 FF C6 66 0C 64 06 65 8B E7 29 A8 1D 65 B9 A6 26 D9 22 77 40 C3 44 68 45 38 FC 23 7C 28 B7 86 8B D0 CD 19 8F EA 31 AF BC 1B FD C9 2A 38 79 B1 6B C3 29 8D EB D5 F0 20 12 64 8C 4D 44 2F AE 7A 61 F6 91 E0 86 ED 8A 4C 75 AF EF 7A DF 6F 13 01 F6 98

Начальный этапный ключ Kb
F8 7E 98 FF C6 66 0C 64 06 65 8B E7 29 A8 1D 65 B9 A6 26 D9 22 77 40 C3 44 68 45 38 FC 23 7C 28

Qo`shHolat			
16	65	89	6C
97	40	EE	AF
6C	D3	09	DD
D9	EC	C1	50
CD	7C	D9	8E
AF	0E	76	04
70	10	B8	AC
11	70	86	3C

1-этап

Этапный ключ Kb[1]:

F8 7E 98 FF C6 66 0C 64 06 65 8B E7 29 A8 1D 65 B9 A6 26 D9 22 77 40 C3 44 68 45 38 FC 23 7C 28

Qo`shBosqichKalit			
EE	1B	11	93
51	26	E2	CB
6A	B6	82	3A
F0	44	DC	35
74	DA	FF	57
8D	79	36	C7
34	78	FD	94
ED	53	FA	14

Aralash			
85	AB	27	85
9B	3C	0A	59
0C	1D	73	FC
33	BA	B1	D2
74	DA	FF	57
8D	79	36	C7
34	78	FD	94
ED	53	FA	14

Sur			
57	ED	78	36
FD	C7	85	53
AB	FA	94	9B
0C	3C	27	14
85	33	1D	0A
73	59	74	BA
DA	B1	FC	8D
34	79	FF	D2

ByteAlmash			
4E	A7	12	7C
99	D4	D0	94
BD	08	C1	8F
48	7D	89	F5
D0	21	CE	AB
10	8E	50	C8
F7	F2	3D	FB
D5	06	CA	F9

2-этап

Этапный ключ Kb[2]:

5F 39 4D 40 EB 2D CD 31 36 C9 13 BA 06 1A 23 42 29 C7 E1 1B E1 45 BC 34 5E 86 68 CC 7F 51 8D 7D

Qo`shBosqichKalit			
11	9E	5F	3C
72	F9	1D	A5
8B	C1	D2	35
4E	67	AA	B7
F9	E6	2F	B0
F1	CB	EC	FC
A9	74	55	37
AA	57	47	84

Aralash			
2B	21	2D	24
BA	62	E5	2B
C3	E0	D2	91
BC	6E	46	CD
F9	E6	2F	B0
F1	CB	EC	FC
A9	74	55	37
AA	57	47	84

Sur			
B0	AA	74	EC
55	FC	2B	57
21	47	37	BA
C3	62	2D	84
24	BC	E0	E5
D2	2B	F9	6E
E6	46	91	F1
A9	CB	2F	CD

ByteAlmash			
DE	F0	50	22
9C	A9	BF	01
67	D3	49	D8
1C	52	E2	34
EB	A8	2E	CC
18	BF	3F	AA
28	92	D7	37
66	CE	7F	41

3-этап

Этапный ключ Kb[3]:

9D D0 30 D1 1A 11 4E 3F 08 DF 0A 2D E1 A2 F4 33 46 63 FA 8C 6B EF 06 FF 72 4A 8E 1E 6C 5A F0 CA

Qo`shBosqichKalit			
43	20	60	F3
86	B8	F1	3E
6F	0C	43	F5
FD	F0	16	07
AD	CB	D4	40
73	50	39	55
5A	D8	59	29
0A	94	8F	8B

Aralash			
8A	86	37	10
7E	8F	DF	66
E6	F4	02	28
BA	68	89	F6
AD	CB	D4	40
73	50	39	55
5A	D8	59	29
0A	94	8F	8B

Sur			
40	0A	D8	39
59	55	8A	94
86	8F	29	7E
E6	8F	37	8B
10	BA	F4	DF
02	66	AD	68
CB	89	28	73
5A	50	D4	F6

ByteAlmash			
81	AB	05	9F
8E	B2	CB	C1
A1	C6	2B	0B
27	C6	1D	8B
B1	C8	60	20
A9	D7	82	42
AF	0A	51	10
5E	6D	80	85

4-этап

Этапный ключ Kb[4]:

51 6F 0D 17 A1 9A 33 1F D4 63 5F 78 37 FB 92 54 70 F3 62 D7 86 53 1B D7 AB E0 40 24 C9 18 9A 88

Qo`shBosqichKalit			
D0	C4	08	88
2F	28	F8	DE
75	A5	74	73
10	3D	8F	DF
C1	3B	02	F7
2F	84	99	95
04	EA	11	34
97	75	1A	0D

Aralash			
A3	C0	7C	34
E5	B2	12	AA
CD	0B	31	37
FB	9B	58	88
C1	3B	02	F7
2F	84	99	95
04	EA	11	34
97	75	1A	0D

Sur			
F7	97	EA	99
11	95	A3	75
C0	1A	34	E5
CD	B2	7C	0D
34	FB	0B	12
31	AA	C1	9B
3B	58	37	2F
04	84	02	88

ByteAlmash			
59	05	8A	6F
CB	25	0E	31
06	11	2F	CC
41	33	7B	9D
2F	13	BD	FD
F6	F0	4D	C1
C0	9E	49	7F
0A	34	0C	F5

5-этап

Этапный ключ Kb[5]:

FB C1 BF DC 92 A3 87 9B 16 BC 32 98 DE BD 5F 02 01 26 48 C4 D4 42 FA E7 A6 1F 69 1E 08 6E D8 A4

Qo`shBosqichKalit			
A2	C4	35	B3
59	86	89	AA
10	AD	1D	54
9F	8E	24	9F
2E	35	F5	39
22	B2	B7	26
66	81	20	61
02	5A	D4	51

Aralash			
E1	CD	E2	6E
82	79	70	BD
9B	79	E8	87
53	A8	30	80
2E	35	F5	39
22	B2	B7	26
66	81	20	61
02	5A	D4	51

Sur			
39	02	81	B7
20	26	E1	5A
CD	D4	61	82
9B	79	E2	51
6E	53	79	70
E8	BD	2E	A8
35	30	87	22
66	B2	F5	80

ByteAlmash			
9F	A9	EF	98
F3	3C	B5	5E
DC	80	A2	C9
8F	06	2D	E8
40	94	06	32
73	F8	E2	BE
1F	30	DD	69
D7	A3	36	75

6-этап

Этапный ключ Kb[6]:

94 C6 F5 EA F8 10 09 32 46 26 A2 17 D7 3D 30 FB 48 F0 43 76 C5 26 3A D7 F7 BD 6F B7 89 80 FB 4C

Qo`shBosqichKalit			
0B	6F	1A	72
0B	2C	BC	6C
9A	A6	00	DE
58	3B	1D	13
08	64	45	44
B6	DE	D8	69
E8	8D	B2	DE
5E	23	CD	39

Aralash			
04	84	77	63
89	CE	8A	74
6C	CF	43	58
62	83	6F	5C
08	64	45	44
B6	DE	D8	69
E8	8D	B2	DE
5E	23	CD	39

Sur			
44	5E	8D	D8
B2	69	04	23
84	CD	DE	89
6C	CE	77	39
63	62	CF	8A
43	74	08	83
64	6F	58	B6
E8	DE	45	5C

ByteAlmash			
24	A4	82	6B
33	AF	0A	3E
34	41	F8	86
A3	6D	03	E0
12	52	07	D5
29	50	4E	C9
AB	B5	9E	3D
94	F8	DA	B3

7-этап

Этапный ключ Kb[7]:

10 BE B9 E9 87 DA 47 82 1B B6 29 31 D6 BF BD EB 7D BC 4C 07 DA 63 E1 FA 63 FF 19 98 31 90 19 96

Qo`shBosqichKalit			
34	1A	3B	82
B4	75	4D	BC
2F	F7	D1	B7
75	D2	BE	0B
6F	EE	4B	D2
F3	33	AF	33
C8	4A	87	A5
A5	68	C3	25

Aralash			
EA	79	57	68
42	2F	2D	9A
59	63	03	31
2D	EE	22	67
6F	EE	4B	D2
F3	33	AF	33
C8	4A	87	A5
A5	68	C3	25

Sur			
D2	A5	4A	AF
87	33	EA	68
79	C3	A5	42
59	2F	57	25
68	2D	63	2D
03	9A	6F	EE
EE	22	31	F3
C8	33	4B	67

ByteAlmash			
F9	E6	45	F0
DD	21	6A	42
06	7E	E6	EA
8E	25	4E	66
42	6E	A4	6E
E5	CF	B0	2F
2F	69	68	0D
8C	21	FC	13

8-этап

Этапный ключ Kb[8]:

49 8E B5 FD EF 5B ED E2 60 3E D3 1F 0F D3 1F F8 CC C1 8C 80 CC B1 7C E5 35 03 AC B7 34 C4 DB 24

Qo`shBosqichKalit			
B0	68	F0	0D
32	7A	87	A0
66	40	35	F5
81	F6	51	9E
8E	AF	28	EE
29	7E	CC	CA
1A	6A	C4	BA
B8	E5	27	37

Aralash			
72	6A	65	68
94	4F	27	CE
81	C3	95	E6
C3	47	B5	2C
8E	AF	28	EE
29	7E	CC	CA
1A	6A	C4	BA
B8	E5	27	37

Sur			
EE	B8	6A	CC
C4	CA	72	E5
6A	27	BA	94
81	4F	65	37
68	C3	C3	27
95	CE	8E	47
AF	B5	E6	29
1A	7E	28	2C

ByteAlmash			
8B	43	A5	AC
4A	10	53	CC
A5	E8	D8	A1
C7	81	C2	49
62	1C	1C	E8
25	6D	95	D3
87	2D	28	76
11	C6	3A	72

so`nggi Bosqich Kalit			
98	F8	7E	98
FF	C6	66	0C
64	06	65	8B
E7	29	A8	1D
65	B9	A6	26
D9	22	77	40
C3	44	68	45
38	FC	23	7C

Qo`shBosqichKalit			
13	BB	DB	34
B5	D6	35	C0
C1	EE	BD	2A
20	A8	6A	54
07	A5	BA	CE
FC	4F	E2	93
44	69	40	33
29	3A	19	0E

Aralash			
13	BB	DB	34
B5	D6	35	C0
C1	EE	BD	2A
20	A8	6A	54
A8	F5	80	C8
32	48	BE	A5
C3	FE	E3	EE
D1	38	6B	4B

********* Результат шифрования**********

13 BB DB 34 B5 D6 35 C0 C1 EE BD 2A 20 A8 6A 54 A8 F5 80 C8 32 48 BE A5 C3 FE E3 EE D1 38 6B 4B

**********Процесс расшифрования**********

Массив замены байтов Bda1

11 C0 06 58 18 D8 79 BB FA BE 89 7E A6 F3 99 D5 73 D0 78 67 D3 3F 45 BF D7 DD 3A 47 6C 37 9F 35 DF
33 2B 60 EF 2F 1B E6 64 0F 90 29 F1 E2 A3 EE 30 62 70 13 F0 1F F5 A1 F7 CF 83 AE 26 FC A7 B8 6E F2
68 DB EB 4A 6B 52 0C A2 88 8C 72 07 57 A9 74 28 4D AC 21 EC 5C 6D 17 93 CA 91 44 1C 5A 96 F4 23 00
A4 7F E3 25 71 31 22 EA 84 04 50 2D F8 E4 24 B4 E8 E0 80 DE 75 DC 38 CC D9 36 3C C3 CE D4 40 AD 9C
43 F6 3E 7B 95 27 1E 8B C8 19 59 9B 3B FB BC 0D 53 B9 D6 97 B7 FD 56 05 9D 41 C1 39 B5 86 61 B2 63
2C 65 ED 15 02 7C 0A D1 12 C2 CB 6F 10 55 08 E5 E1 C4 9E 5B 0E 46 A0 4E AB A8 49 B6 94 4C 6A 1A E9
8F 7A BA 82 FF 8A 3D 92 1D 9A 85 2A F9 AA C7 34 5E 66 5D 69 C5 C9 CD 87 54 76 4F 32 2E 0B B0 03 A5
48 51 01 42 09 98 8E FE 81 AF E7 B1 20 B3 14 5F DA BD D2 77 8D 4B 16 C6 7D

Массив замены байтов Bda2

22 57 67 77 42 97 C0 CF 06 F2 04 C5 02 C7 A3 36 CA 1A 63 FB 1E 07 A6 13 D2 F5 B4 F3 C3 D7 82 18 35
DB EC 54 44 95 16 14 E6 43 E4 3C E2 B5 E0 34 AC 75 DC B2 84 A8 25 F1 05 D5 28 D3 F6 B6 23 F9 A2 CD
66 B8 FE 93 26 01 C6 37 C4 D6 0C C1 08 32 74 52 62 72 D4 92 B9 17 B7 F7 E7 15 4B FD EF 98 41 51 68 1D
FA 1F A9 21 03 D0 A5 D8 C8 CE A1 99 9F 50 2C 9C 80 90 29 19 BF 33 DF 7C FF 0F 38 2F 3F 4F 8D 3D 4A
8F 89 AF 0A 86 EA EE 4C E3 48 2E 8B 6D 46 5B E8 8E 9E AE BD 76 EB 56 55 0D 58 70 5A 94 7B 6C 5E
6A 60 1C BC FC 6E 64 CC C2 AD 69 5F 6B 61 5C 4D 6F 59 71 BB 73 87 DD 53 0B 1B 2B 3B 9B 65 7D 4E
3E 7E 81 A4 83 DA 11 E5 BE CB 40 3A 7F 8C 47 7A 8A 9A 91 BA C9 45 E9 96 09 B0 A0 39 49 2D 9D 79
2A 30 0E 27 00 F0 24 F8 A7 D9 20 AA E1 85 5D 78 88 31 B1 DE B3 F4 AB D1 12 ED 10

Shakl seans Kalit Kse

98 F8 7E 98 FF C6 66 0C 64 06 65 8B E7 29 A8 1D 65 B9 A6 26 D9 22 77 40 C3 44 68 45 38 FC 23 7C 28 B7
86 8B D0 CD 19 8F EA 31 AF BC 1B FD C9 2A 38 79 B1 6B C3 29 8D EB D5 F0 20 12 64 8C 4D 44 2F AE
7A 61 F6 91 E0 86 ED 8A 4C 75 AF EF 7A DF 6F 13 01 F6

Начальный этапный ключ Kb

98 F8 7E 98 FF C6 66 0C 64 06 65 8B E7 29 A8 1D 65 B9 A6 26 D9 22 77 40 C3 44 68 45 38 FC 23 7C

Aralash			
13	BB	DB	34
B5	D6	35	C0
C1	EE	BD	2A
20	A8	6A	54
07	A5	BA	CE
FC	4F	E2	93
44	69	40	33
29	3A	19	0E

boshlang`ich Bosqich Kalit			
98	F8	7E	98
FF	C6	66	0C
64	06	65	8B
E7	29	A8	1D
65	B9	A6	26
D9	22	77	40
C3	44	68	45
38	FC	23	7C

Qo`shBosqichKalit			
8B	43	A5	AC
4A	10	53	CC
A5	E8	D8	A1
C7	81	C2	49
62	1C	1C	E8
25	6D	95	D3
87	2D	28	76
11	C6	3A	72

1-этап

Этапный ключ Kb[1]:

98 F8 7E 98 FF C6 66 0C 64 06 65 8B E7 29 A8 1D 65 B9 A6 26 D9 22 77 40 C3 44 68 45 38 FC 23 7C

ByteAlmash			
EE	B8	6A	CC
C4	CA	72	E5
6A	27	BA	94
81	4F	65	37
68	C3	C3	27
95	CE	8E	47
AF	B5	E6	29
1A	7E	28	2C

Sur			
72	6A	65	68
94	4F	27	CE
81	C3	95	E6
C3	47	B5	2C
8E	AF	28	EE
29	7E	CC	CA
1A	6A	C4	BA
B8	E5	27	37

Aralash			
B0	68	F0	0D
32	7A	87	A0
66	40	35	F5
81	F6	51	9E
8E	AF	28	EE
29	7E	CC	CA
1A	6A	C4	BA
B8	E5	27	37

Qo`shBosqichKalit			
F9	E6	45	F0
DD	21	6A	42
06	7E	E6	EA
8E	25	4E	66
42	6E	A4	6E
E5	CF	B0	2F
2F	69	68	0D
8C	21	FC	13

2-этап

Этапный ключ Kb[2]:

49 8E B5 FD EF 5B ED E2 60 3E D3 1F 0F D3 1F F8 CC C1 8C 80 CC B1 7C E5 35 03 AC B7 34 C4 DB 24

ByteAlmash			
D2	A5	4A	AF
87	33	EA	68
79	C3	A5	42
59	2F	57	25
68	2D	63	2D
03	9A	6F	EE
EE	22	31	F3
C8	33	4B	67

Sur			
EA	79	57	68
42	2F	2D	9A
59	63	03	31
2D	EE	22	67
6F	EE	4B	D2
F3	33	AF	33
C8	4A	87	A5
A5	68	C3	25

Aralash			
34	1A	3B	82
B4	75	4D	BC
2F	F7	D1	B7
75	D2	BE	0B
6F	EE	4B	D2
F3	33	AF	33
C8	4A	87	A5
A5	68	C3	25

Qo`shBosqichKalit			
24	A4	82	6B
33	AF	0A	3E
34	41	F8	86
A3	6D	03	E0
12	52	07	D5
29	50	4E	C9
AB	B5	9E	3D
94	F8	DA	B3

3-этап

Этапный ключ Kb[3]:

10 BE B9 E9 87 DA 47 82 1B B6 29 31 D6 BF BD EB 7D BC 4C 07 DA 63 E1 FA 63 FF 19 98 31 90 19 96

ByteAlmash			
44	5E	8D	D8
B2	69	04	23
84	CD	DE	89
6C	CE	77	39
63	62	CF	8A
43	74	08	83
64	6F	58	B6
E8	DE	45	5C

Sur			
04	84	77	63
89	CE	8A	74
6C	CF	43	58
62	83	6F	5C
08	64	45	44
B6	DE	D8	69
E8	8D	B2	DE
5E	23	CD	39

Aralash			
0B	6F	1A	72
0B	2C	BC	6C
9A	A6	00	DE
58	3B	1D	13
08	64	45	44
B6	DE	D8	69
E8	8D	B2	DE
5E	23	CD	39

Qo`shBosqichKalit			
9F	A9	EF	98
F3	3C	B5	5E
DC	80	A2	C9
8F	06	2D	E8
40	94	06	32
73	F8	E2	BE
1F	30	DD	69
D7	A3	36	75

4-этап

Этапный ключ Kb[4]:

94 C6 F5 EA F8 10 09 32 46 26 A2 17 D7 3D 30 FB 48 F0 43 76 C5 26 3A D7 F7 BD 6F B7 89 80 FB 4C

ByteAlmash			
39	02	81	B7
20	26	E1	5A
CD	D4	61	82
9B	79	E2	51
6E	53	79	70
E8	BD	2E	A8
35	30	87	22
66	B2	F5	80

Sur			
E1	CD	E2	6E
82	79	70	BD
9B	79	E8	87
53	A8	30	80
2E	35	F5	39
22	B2	B7	26
66	81	20	61
02	5A	D4	51

Aralash			
A2	C4	35	B3
59	86	89	AA
10	AD	1D	54
9F	8E	24	9F
2E	35	F5	39
22	B2	B7	26
66	81	20	61
02	5A	D4	51

Qo`shBosqichKalit			
59	05	8A	6F
CB	25	0E	31
06	11	2F	CC
41	33	7B	9D
2F	13	BD	FD
F6	F0	4D	C1
C0	9E	49	7F
0A	34	0C	F5

5-этап

Этапный ключ Kb[5]:

FB C1 BF DC 92 A3 87 9B 16 BC 32 98 DE BD 5F 02 01 26 48 C4 D4 42 FA E7 A6 1F 69 1E 08 6E D8 A4

ByteAlmash			
F7	97	EA	99
11	95	A3	75
C0	1A	34	E5
CD	B2	7C	0D
34	FB	0B	12
31	AA	C1	9B
3B	58	37	2F
04	84	02	88

Sur			
A3	C0	7C	34
E5	B2	12	AA
CD	0B	31	37
FB	9B	58	88
C1	3B	02	F7
2F	84	99	95
04	EA	11	34
97	75	1A	0D

Aralash			
D0	C4	08	88
2F	28	F8	DE
75	A5	74	73
10	3D	8F	DF
C1	3B	02	F7
2F	84	99	95
04	EA	11	34
97	75	1A	0D

Qo`shBosqichKalit			
81	AB	05	9F
8E	B2	CB	C1
A1	C6	2B	0B
27	C6	1D	8B
B1	C8	60	20
A9	D7	82	42
AF	0A	51	10
5E	6D	80	85

6-этап

Этапный ключ Kb[6]:

51 6F 0D 17 A1 9A 33 1F D4 63 5F 78 37 FB 92 54 70 F3 62 D7 86 53 1B D7 AB E0 40 24 C9 18 9A 88

ByteAlmash			
40	0A	D8	39
59	55	8A	94
86	8F	29	7E
E6	8F	37	8B
10	BA	F4	DF
02	66	AD	68
CB	89	28	73
5A	50	D4	F6

Sur			
8A	86	37	10
7E	8F	DF	66
E6	F4	02	28
BA	68	89	F6
AD	CB	D4	40
73	50	39	55
5A	D8	59	29
0A	94	8F	8B

Aralash			
43	20	60	F3
86	B8	F1	3E
6F	0C	43	F5
FD	F0	16	07
AD	CB	D4	40
73	50	39	55
5A	D8	59	29
0A	94	8F	8B

Qo`shBosqichKalit			
DE	F0	50	22
9C	A9	BF	01
67	D3	49	D8
1C	52	E2	34
EB	A8	2E	CC
18	BF	3F	AA
28	92	D7	37
66	CE	7F	41

7-этап

Этапный ключ Kb[7]:

9D D0 30 D1 1A 11 4E 3F 08 DF 0A 2D E1 A2 F4 33 46 63 FA 8C 6B EF 06 FF 72 4A 8E 1E 6C 5A F0 CA

ByteAlmash			
B0	AA	74	EC
55	FC	2B	57
21	47	37	BA
C3	62	2D	84
24	BC	E0	E5
D2	2B	F9	6E
E6	46	91	F1
A9	CB	2F	CD

Sur			
2B	21	2D	24
BA	62	E5	2B
C3	E0	D2	91
BC	6E	46	CD
F9	E6	2F	B0
F1	CB	EC	FC
A9	74	55	37
AA	57	47	84

Aralash			
11	9E	5F	3C
72	F9	1D	A5
8B	C1	D2	35
4E	67	AA	B7
F9	E6	2F	B0
F1	CB	EC	FC
A9	74	55	37
AA	57	47	84

Qo`shBosqichKalit			
4E	A7	12	7C
99	D4	D0	94
BD	08	C1	8F
48	7D	89	F5
D0	21	CE	AB
10	8E	50	C8
F7	F2	3D	FB
D5	06	CA	F9

8-этап

Этапный ключ Kb[8]:

5F 39 4D 40 EB 2D CD 31 36 C9 13 BA 06 1A 23 42 29 C7 E1 1B E1 45 BC 34 5E 86 68 CC 7F 51 8D 7D

ByteAlmash			
57	ED	78	36
FD	C7	85	53
AB	FA	94	9B
0C	3C	27	14
85	33	1D	0A
73	59	74	BA
DA	B1	FC	8D
34	79	FF	D2

Sur			
85	AB	27	85
9B	3C	0A	59
0C	1D	73	FC
33	BA	B1	D2
74	DA	FF	57
8D	79	36	C7
34	78	FD	94
ED	53	FA	14

Aralash			
EE	1B	11	93
51	26	E2	CB
6A	B6	82	3A
F0	44	DC	35
74	DA	FF	57
8D	79	36	C7
34	78	FD	94
ED	53	FA	14

Qo`shBosqichKalit			
16	65	89	6C
97	40	EE	AF
6C	D3	09	DD
D9	EC	C1	50
CD	7C	D9	8E
AF	0E	76	04
70	10	B8	AC
11	70	86	3C

Qo`shHolat			
30	31	32	33
34	35	36	37
38	39	41	42
43	44	45	46
30	31	32	33
34	35	36	37
38	39	41	42
43	44	45	46

******* Результат расшифрования*******

30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46

УДК: ОКС 35.040 Группа П85

Ключевые слова: алгоритм шифрования данных, шифртекст, зашифрования, расшифрование, ключ, сеансовый ключ, функциональный ключ

Заместиетль директора ГУП «UNICON.UZ» Х.П. Хасанов

Главный научнный сотрудник ГУП «UNICON.UZ» д.т.н., проф. П.Ф. Хасанов

Начальник Научно-исследовательского отдела средств защиты информации О.Х. Расулов

Нальник Научно-исследовательского отдела криптографии к.т.н. О.П. Ахмедова

Заместитель начальника Научно-исследовательского отдела криптографии Ж.Д. Мукимов

Начальник лаборатории криптоанализа А.Б. Давлатов

Нормоконтроль Н. К. Травина

СОГЛАСОВАНО

Служба национальной безопасности Республики Узбекистан

Письмо от 27 марта 2007 года №39/657