Криптографические алгоритмы и IT платформы «ECLECTIC-DT»

THE THEORY OF PLAFALES:
Криптографический комплекс «ECLECTIC-DT-1»

Дмитрий Топчий (Dmytro Topchyi)

2014

СОДЕРЖАНИЕ

СТРУКТУРНАЯ ЧАСТЬ

Согласно принципа Кирхгофа: секретность шифра обеспечивается секретностью ключа, а не секретностью алгоритма шифрования. Противник имеет всю информацию о применяемом криптоалгоритме, ему неизвестен только реально использованный ключ [1].

Криптографические (математические) алгоритмы серии «ECLECTIC-DT» основываются на работе «The theory of plafales: the proof algorithms for millennium problems» [2, 3, 4], а также на материал XLII конференции по прикладной математике (Польская Академия Наук) [5, 6].

Предлагаемые к рассмотрению и внедрению криптографические (математические) алгоритмы серии «ECLECTIC-DT», а также IT платформы которые могут быть созданы на базе указанной серии, преследуют решения следующих задач: **1.** Криптографический комплекс «ECLECTIC-DT-1»: система «Friend-Foe»; **2.** Ввод стандарта шифрования данных – «ECLECTIC-DT-2»; **3.** Ввод нового стандарта «ECLECTIC-DT-3», определяющего алгоритм и процедуру вычисления хеш-функции; **4.** Комплекс «ECLECTIC-DT-4»: построение генераторов псевдослучайных чисел; **5.** IT платформа (комплекс) «ECLECTIC-DT-5»: реализация равенства классов сложности P=NP (разрушение криптосистем с открытым ключом); **6.** Создание платформы «ECLECTIC-DT-6»: конструирование базисных функций в автоматическом режиме в методе конечных элементов (МКЭ).

Представленные алгоритмы серии «ECLECTIC-DT» несут теоретический характер. Для дальнейшей реализации в программно-аппаратном комплексе необходимы следующие шаги: **1.** Создание софта под отдельно взятый алгоритм (в индивидуальном порядке); **2.** Проведение криптоанализа; **3.** Пусконаладка и шеф-монтаж (инсталлирование).

ГЛАВА 1

«ECLECTIC-DT-1»

1.1 Введение и терминология

«ECLECTIC-DT-1» представляет собой алгоритм итерационного блочного симметричного шифрования 128-битных блоков данных ключом 256 бит. Количество раундов – 14.

Байт – последовательность из 8 битов. В контексте данного алгоритма байт рассматривается как plafal (плафал) – PF.

Блок – последовательность из 16 байтов, над которой оперирует алгоритм. В контексте данного алгоритма блок рассматривается как „docking“ of plafales (процедура стыковки плафалов) – $\mathrm{PF}_{S^{16}}^{doc}$.

Матрица состояний – комплекс из 16 байтов, отображающий состояние блока и формы перед, в ходе, и после выполнения всех раундовых процедур.

Форма – последовательность из 16 байтов, которая рассматривается как plafal (плафал) – PF_{ad}^{uniq}.

Ключ – последовательность из 32 байтов, используемая в качестве ключа шифрования.

Раунд – итерация цикла преобразований над матрицей состояний. Количество раундов – 14.

Ключ раунда – ключ, применяемый в раунде. Вычисляется для каждого раунда.

Для зашифрования в алгоритме «ECLECTIC-DT-1» применяются следующие процедуры преобразования данных:

SubBytes – подстановка байтов в матрице состояний с помощью таблицы подстановок.

PerBits – перестановка бит в байте. В контексте данного алгоритма рассматривается как поворот против часовой стрелки absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} вокруг центра симметрии на угол $\varphi = \frac{360^\circ \cdot n}{8} = 45^\circ \cdot n$, n – количество поворотов.

ShiftBytes – циклический сдвиг байт в матрице состояний на различные ве-

личины. В контексте данного алгоритма рассматривается как параллельный перенос PF_i на PF_j; $i \neq j$ комплекса $\mathrm{PF}_{S^{16}}^{doc}$.

PerBytes – перестановка байт в матрице состояний. В контексте данного алгоритма рассматривается как поворот против часовой стрелки absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} вокруг центра симметрии на угол $\varphi = \frac{360^\circ \cdot n}{16} = 22.5^\circ \cdot n$, n – количество поворотов.

Последовательность выполнения процедур с 1-13 раунды:

SubBytes→PerBits→ShiftBytes→PerBytes

Последовательность выполнения процедур в 14 раунде:

SubBytes→PerBits→ShiftBytes

Для расшифрования в алгоритме «ECLECTIC-DT-1» применяются следующие процедуры преобразования данных:

InvSubBytes – подстановка байтов в матрице состояний с помощью обратной таблицы подстановок.

InvPerBits – перестановка бит в байте. В контексте данного алгоритма рассматривается как поворот вокруг часовой стрелки absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} вокруг центра симметрии на угол $\varphi = \frac{360^\circ \cdot n}{8} = 45^\circ \cdot n$, n – количество поворотов.

InvShiftBytes – циклический сдвиг байт в матрице состояний на различные величины. В контексте данного алгоритма рассматривается как параллельный перенос PF_j на PF_i; $j \neq i$ комплекса $\mathrm{PF}_{S^{16}}^{doc}$.

InvPerBytes – перестановка байт в матрице состояний. В контексте данного алгоритма рассматривается как поворот вокруг часовой стрелки absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} вокруг центра симметрии на угол $\varphi = \frac{360^\circ \cdot n}{16} = 22.5^\circ \cdot n$, n – количество поворотов.

Последовательность выполнения процедур в 1 раунде:

InvShiftBytes→InvPerBits→InvSubBytes

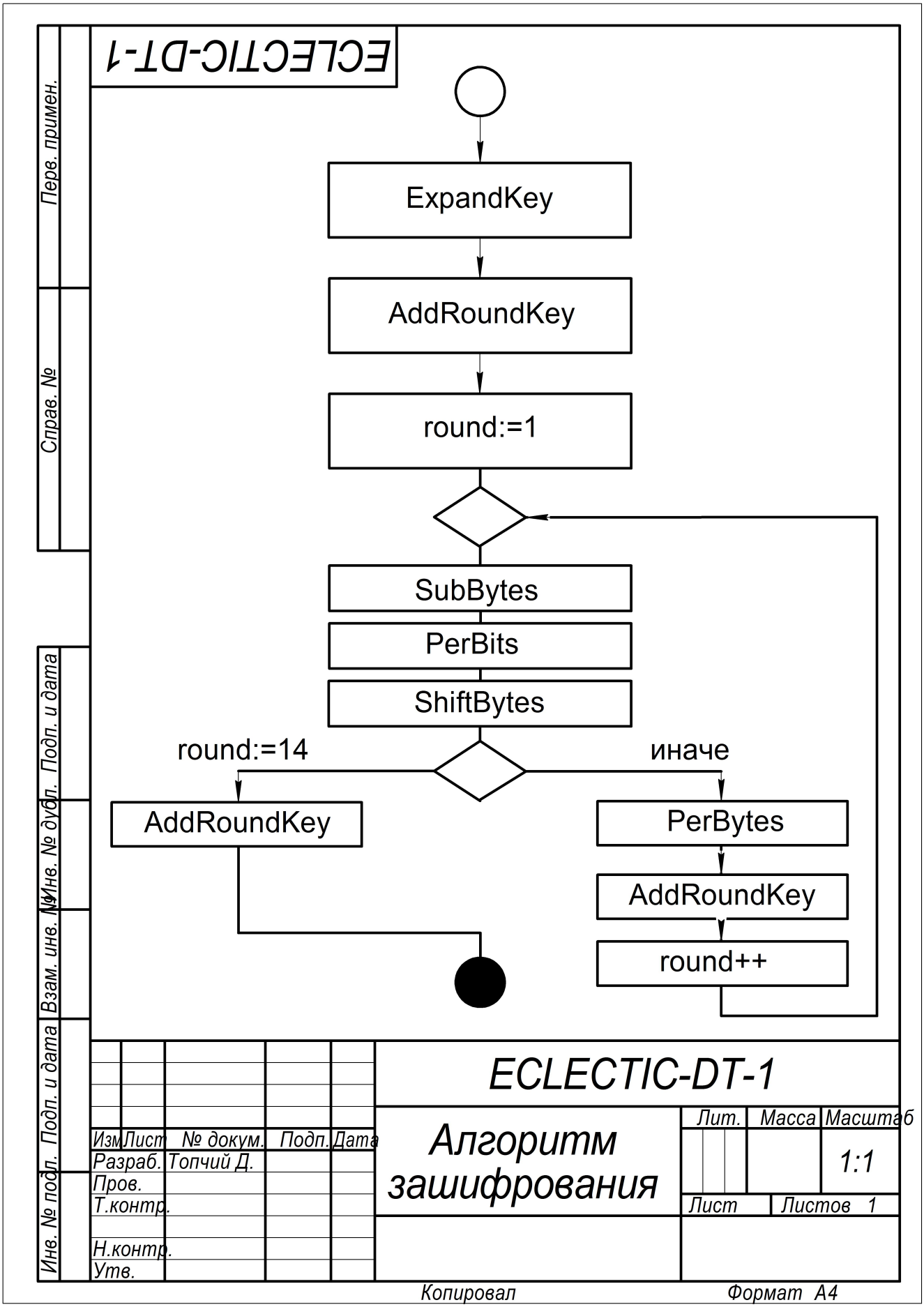
Последовательность выполнения процедур с 2-14 раунды:

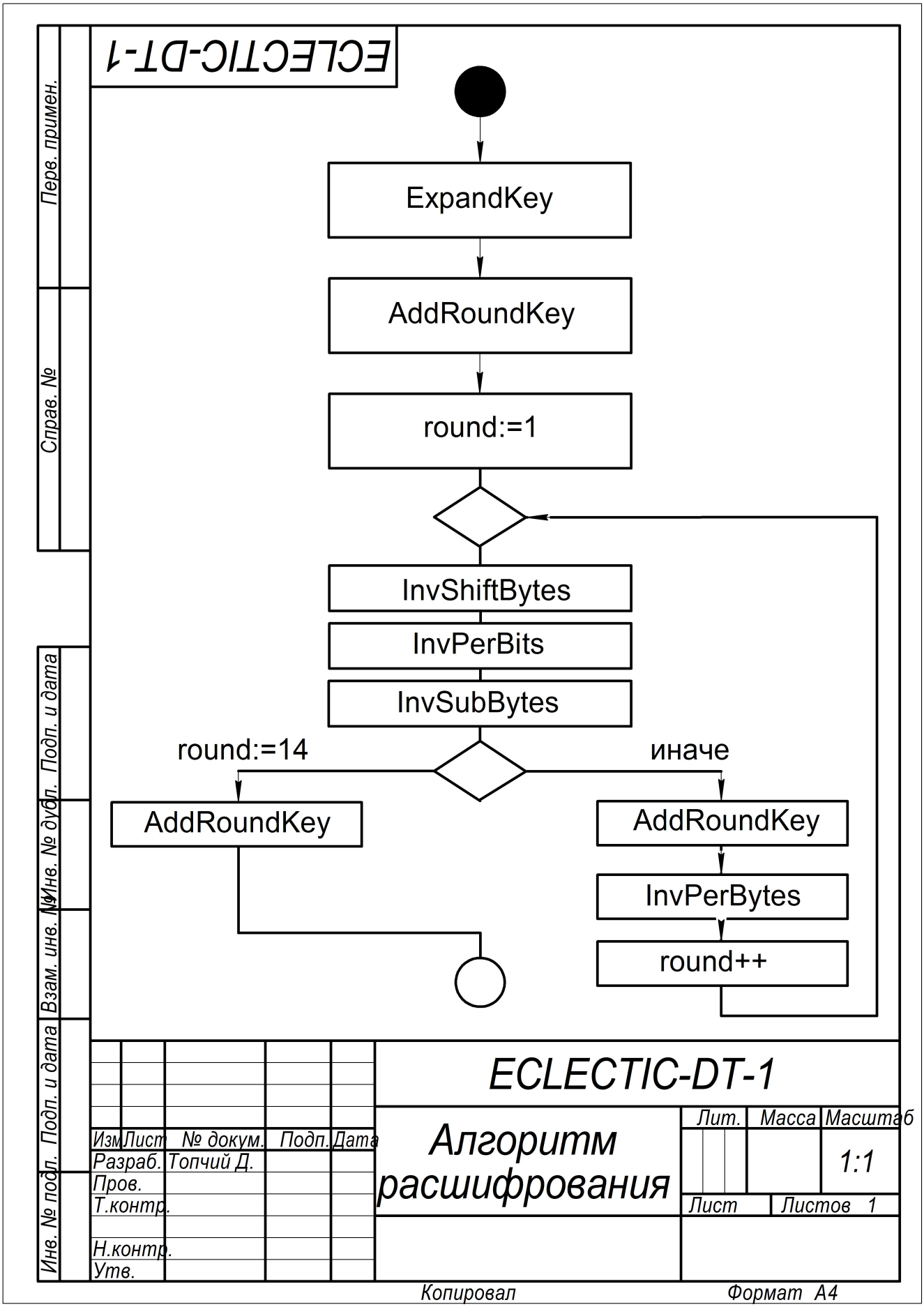
InvPerBytes→InvShiftBytes→InvPerBits→InvSubBytes

Алгоритм выработки ключей (Key Schedule)

ExpandKey – вычисление раундных ключей для всех раундов.

AddRoundKey – сложение ключа раунда с матрицей состояния.


ECLECTIC-DT-1
ExpandKey
AddRoundKey
round:=1
SubBytes
PerBits
ShiftBytes
round:=14
иначе
AddRoundKey
PerBytes
AddRoundKey
round++
Перв. примен.
Справ. №
Подп. и дата
Инв. № дубл.
Взам. инв. №
Подп. и дата
Инв. № подл.
ECLECTIC-DT-1
Алгоритм зашифрования
Изм. Лист № докум. Подп. Дата
Разраб. Топчий Д.
Пров.
Т.контр.
Н.контр.
Утв.
Лит. Масса Масштаб
1:1
Лист Листов 1
Копировал
Формат А4


ECLECTIC-DT-1
ExpandKey
AddRoundKey
round:=1
InvShiftBytes
InvPerBits
InvSubBytes
round:=14
иначе
AddRoundKey
AddRoundKey
InvPerBytes
round++
Перв. примен.
Справ. №
Подп. и дата
Инв. № дубл.
Взам. инв. №
Подп. и дата
Инв. № подл.
ECLECTIC-DT-1
Алгоритм расшифрования
Изм. Лист № докум. Подп. Дата
Разраб. Топчий Д.
Пров.
Т.контр.
Н.контр.
Утв.
Лит. Масса Масштаб
1:1
Лист Листов 1
Копировал
Формат А4

1.2 Математические основы алгоритма

1.2.1 Байт

В криптоалгоритме операции выполняются над байтами. Количество plafales (плафалов) равно количеству состояний, то есть 2^8=256. Согласно [4, с 16] для байта $\{b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0\}$ существует взаимно-однозначное соответствие (биекция), которое образует plafal (плафал): $b_{8-i} \leftrightarrow i, i = \overline{1;8}$; где i – сторона правильного 8-угольника. Например, для байта 00010011 соответствующий plafal (плафал) – рис.1.1:

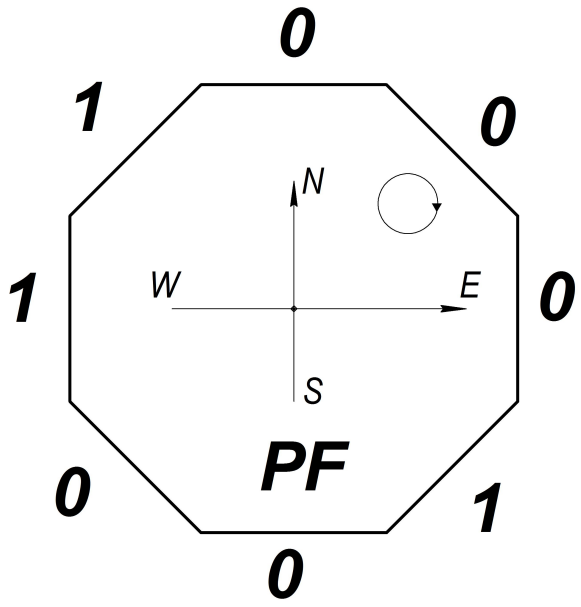


Рисунок 1.1 — Представление байта 00010011

Ориентация сторон plafal (плафала) с севера на северо-запад, то есть: 1 сторона находится на севере, 2 сторона находится на северо-востоке и т.д. Соответственно, правило обхода – вокруг часовой стрелки. При поворотах вокруг центра симметрии (операции PerBits и InvPerBits) plafal (плафала) ориентация сторон не изменяется.

1.2.2 Блок

Блок представляет собой последовательность из 16 plafales (плафалов), фактически образует сотовую структуру – „docking“ of plafales (стыковка плафалов) [4, с 605] из шестнадцати plafales (плафалов) PF_{ad}^{uniq} [4, с 589] – рис.1.2. Каждый из plafales (плафалов) имеет между собой 2, 3 или 4 общие стороны. На рис.1.3 представлена сотовая структура **позиций** комплекса „docking“ of plafales. То есть, $\mathrm{PF}_k, k = \overline{1;16}$ означает, что один из шестнадцати plafales (плафалов) занимает k-ю позицию в сотовой структуре; при этом PF_k не является k-м плафалом (*в алгоритме не существует понятие k-ого плафала*). Исходя из определения процедуры „docking“ of plafales, имеем следующее: plafal (плафал), который имеет 2, 3 или 4 общие стороны с другими plafales (плафалами), безусловно, сохраняет свою байтовую структуру (на рис.1.2: 3-ей стороне PF_1 (плафал который занимает 1-ю позицию) соответствует множество $\{0\}$ и 7-ой стороне PF_5 (плафал который занимает 5-ю позицию) – множество $\{1\}$).

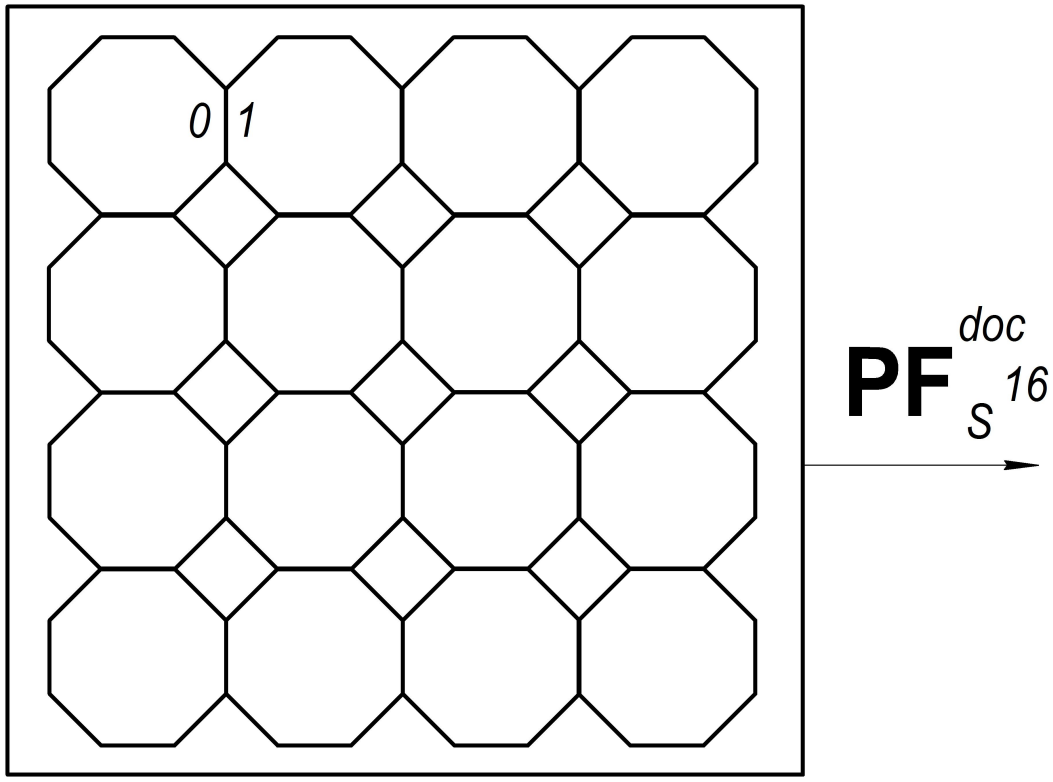


Рисунок 1.2 — Сотовая структура блока („docking“ of plafales)

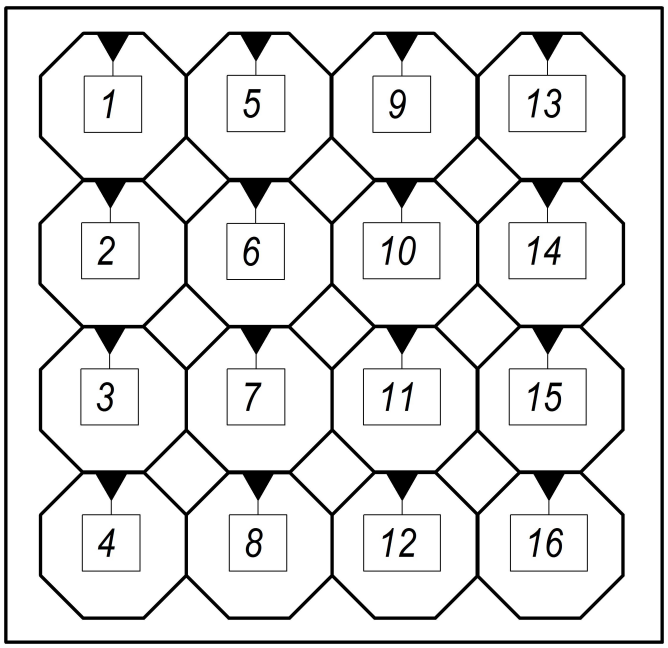


Рисунок 1.3 — Сотовая структура позиций блока

1.2.3 The static canvas of plafal (статический ковер плафала)

Вышеописанная сотовая структура блока находится на the static canvas of plafal (статическом ковре плафала) – PF_{16}^{USP} [4, с 16], который представляет собой плоскость в геометрии – двумерное пространство нулевой кривизны (то есть на E_2). В контексте алгоритма, сотовую структуру блока и аффинные преобразования совершаемые над ним (раундовые операции PerBits, InvPerBits, ShiftBytes, InvShiftBytes) будем производить над правильными 8-угольниками, образующие каждый из plafales (плафалов) комплекса $\mathrm{PF}_{S^{16}}^{doc}$. Раундовые операции PerBytes и InvPerBytes будем производить над правильным 16-угольником. Все шестнадцать правильных 8-угольников конгруэнтны между собой. Характеристики правильного 8-угольника: радиус вписанной окружности $r = 1$, длина стороны $a = \frac{2}{1+\sqrt{2}}$, радиус описанной окружности $R = \sqrt[4]{\frac{8}{3+2\sqrt{2}}}$. Соответственно, для комплекса $\mathrm{PF}_{S^{16}}^{doc}$: начало правой прямоугольной системы координат – т. O_1 (0;0) находится в центре симметрии PF_1, соответствующий репер – R_1 (O_1,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$); вектора $\overrightarrow{e_1}$ и $\overrightarrow{e_2}$ являются ортонормированными (с еденичными длинами). Координаты центров симметрий $\mathrm{PF}_k, k = \overline{1;16}$ и соответствующие реперы:

Таблица 1.1

PF_k	Координаты центров симметрий	Репер
PF_1	т. O_1 (0;0)	R_1 (O_1,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_2	т. O_2 (0;-2)	R_2 (O_2,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_3	т. O_3 (0;-4)	R_3 (O_3,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_4	т. O_4 (0;-6)	R_4 (O_4,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_5	т. O_5 (2;0)	R_5 (O_5,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_6	т. O_6 (2;-2)	R_6 (O_6,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_7	т. O_7 (2;-4)	R_7 (O_7,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_8	т. O_8 (2;-6)	R_8 (O_8,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_9	т. O_9 (4;0)	R_9 (O_9,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{10}	т. O_{10} (4;-2)	R_{10} (O_{10},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{11}	т. O_{11} (4;-4)	R_{11} (O_{11},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{12}	т. O_{12} (4;-6)	R_{12} (O_{12},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{13}	т. O_{13} (6;0)	R_{13} (O_{13},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{14}	т. O_{14} (6;-2)	R_{14} (O_{14},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{15}	т. O_{15} (6;-4)	R_{15} (O_{15},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)
PF_{16}	т. O_{16} (6;-6)	R_{16} (O_{16},$\overrightarrow{e_1}$,$\overrightarrow{e_2}$)

Вышеописанная конфигурация показана на рис.1.4.

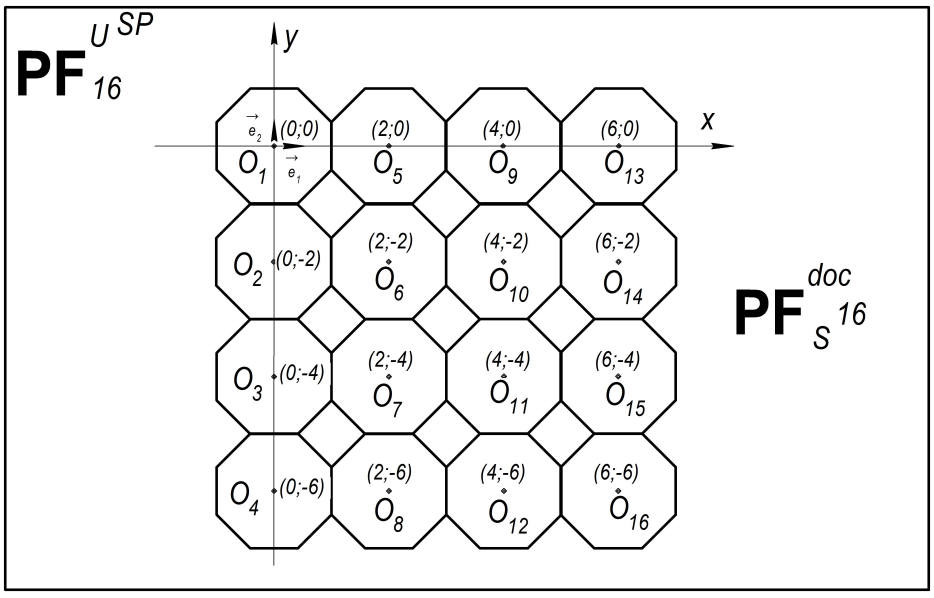


Рисунок 1.4 — Конфигурация на E_2

Каждый из шестнадцати правильных 8-угольников однозначно определяется благодаря координатам своих вершин. Достаточно рассмотреть положение PF_1 – рис.1.5; координаты вершин остальных правильных 8-угольников определяются вектором параллельного переноса $\overrightarrow{O_1O_j}, j = \overline{2;16}$.

Координаты вершин правильного 8-угольника и правильного 16-угольника определяются формулами:

$$x_i = O_x + R \cdot \cos(\phi_0 + \frac{2\pi i}{8}), i = \overline{0; n-1}$$
$$y_i = O_y + R \cdot \sin(\phi_0 + \frac{2\pi i}{8}), i = \overline{0; n-1}$$

$(O_x; O_y)$ – координаты центра симметрии; ϕ_0 – угловая координата первой вершины.

Координаты вершин PF_1: (≈0.415;1), (1;≈0.415), (1;≈-0.415), (≈0.415;-1), (≈-0.415;-1), (-1;≈-0.415), (-1;≈0.415), (≈-0.415;1).

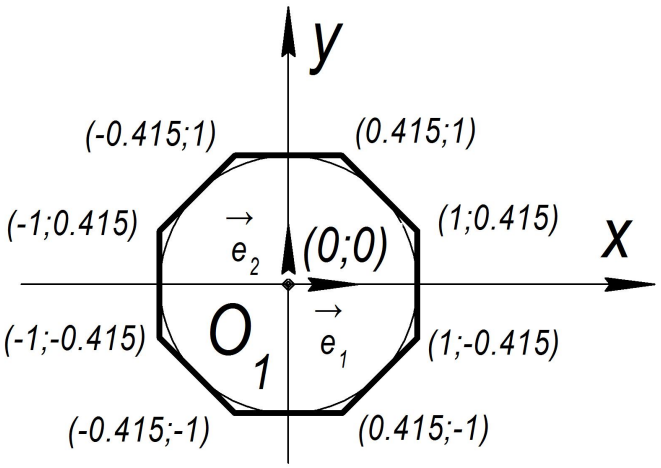


Рисунок 1.5 — Координаты вершин PF_1

Северная сторона PF_1, до и после выполнения раундовых процедур, определяется в репере R_1 координатами двух вершин: (≈0.415;1), (≈-0.415;1). Ориентация остальных сторон определяется согласно п.1.2.1.

1.2.4 Матрица состояний

Матрица состояний (рис.1.6, рис.1.7) отображает состояние блока и формы в ходе выполнения всех раундовых процедур. То есть матрица 4×4: $A = [a_{ij}|_{i=\overline{1;4}}^{j=\overline{1;4}}]$

отражает неизменный порядок байтов в блоке по следующему принципу:

$$PF_k \leftrightarrow a_{ij}, k = \overline{1;16} \tag{1.1}$$

Таблица 1.2

$\text{PF}_1 \leftrightarrow a_{11}$	$\text{PF}_2 \leftrightarrow a_{21}$	$\text{PF}_3 \leftrightarrow a_{31}$	$\text{PF}_4 \leftrightarrow a_{41}$
$\text{PF}_5 \leftrightarrow a_{12}$	$\text{PF}_6 \leftrightarrow a_{22}$	$\text{PF}_7 \leftrightarrow a_{32}$	$\text{PF}_8 \leftrightarrow a_{42}$
$\text{PF}_9 \leftrightarrow a_{13}$	$\text{PF}_{10} \leftrightarrow a_{23}$	$\text{PF}_{11} \leftrightarrow a_{33}$	$\text{PF}_{12} \leftrightarrow a_{43}$
$\text{PF}_{13} \leftrightarrow a_{14}$	$\text{PF}_{14} \leftrightarrow a_{24}$	$\text{PF}_{15} \leftrightarrow a_{34}$	$\text{PF}_{16} \leftrightarrow a_{44}$

$$\Lambda_{16} \leftrightarrow a_{ij} \tag{1.2}$$

PF_k – сотовая структура позиций комплекса „docking“ of plafales (рис.1.3). Поясним подробнее: a_{11} – 1-ый байт, a_{21} – 2-ой байт и т.д. В ходе перестановок плафалов (ShiftBytes, InvShiftBytes, PerBytes, InvPerBytes) на позиции a_{11} (1-ого байта) может оказаться PF_9 и т.п. Так как каждому байту $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ записанному в двоичной системе счисления соответствует запись в шестнадцатеричной системе счисления – Λ_{16}, то согласно соответствию (1.2), в матрице состояний первому байту a_{11} будет соответствовать запись байта в Λ_{16}, которая снимается с PF_1, согласно ориентации сторон правильного 8-угольника (п.1.2.1, п.1.2.3). Безусловно, с Λ_{16} a_{ij} однозначно восстанавливается PF_k. Например, для байта 00010011 (п.1.2.1): $\{00010011\} \leftrightarrow 13_{16}$. Соответственно, $13_{16} \leftrightarrow a_{11}$ (в случае PF_1).

a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}

Рисунок 1.6 — Матрица состояний (общий вид)

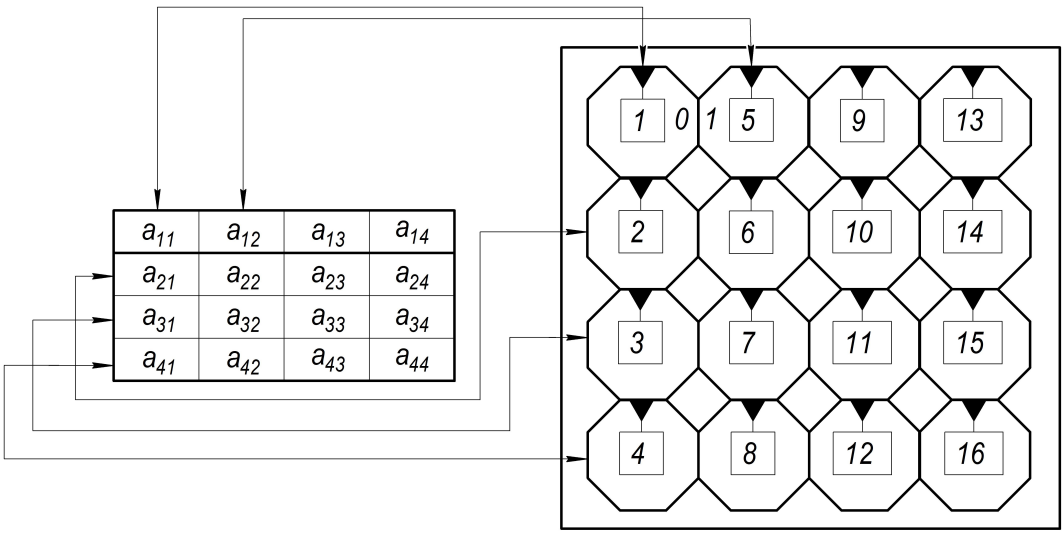


Рисунок 1.7 — Зависимость матрицы состояний и структуры позиций блока

1.2.5 Форма

Форма представляет собой plafal (плафал) – PF_{ad}^{uniq} [4, с 16, с 589] образованный следующем образом: для правильного шестнадцатиугольника существует взаимно-однозначное соответствие, которое образует plafal (плафал): $a_{ij} \leftrightarrow l \Leftrightarrow \Lambda_{16} \leftrightarrow l, l = \overline{1;16}$; где l – сторона правильного 16-угольника. Например, для матрицы состояний (рис.1.8) соответствующий plafal (плафал) – рис.1.9:

a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}

↔

13	17	AB	64
57	FD	E3	CA
83	B7	EA	48
D4	C3	BF	A1

Рисунок 1.8 — Матрица состояний (в качестве примера)

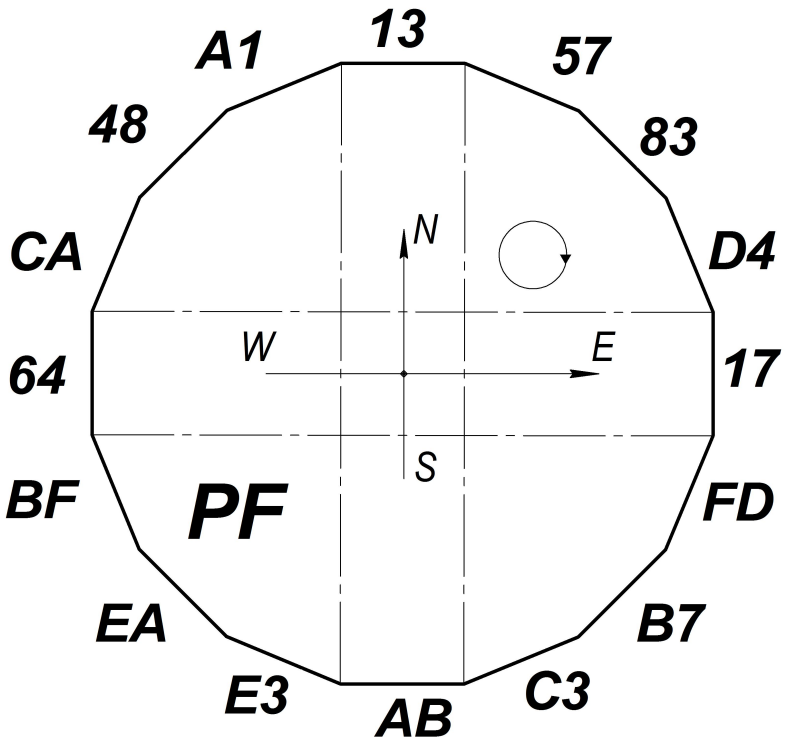


Рисунок 1.9 — Форма

Ориентация сторон plafal (плафала) с севера на севернее северо-запада, то есть: 1 сторона находится на севере, 2 сторона находится на севернее северо-востока и т.д. Соответственно, правило обхода – вокруг часовой стрелки. При поворотах вокруг центра симметрии (операции PerBytes и InvPerBytes) plafal (плафала) ориентация сторон не изменяется. Раундовые операции PerBytes и InvPerBytes будем производить над правильным 16-угольником (согласно п.1.2.3). Правильный 16-угольник находится на статическом ковре плафала, отличном от того на котором находится комплекс PF^{doc}_{S16}. Характеристики правильного 16-угольника: радиус вписанной окружности $r = 1$, длина стороны $a = 2r \tan \frac{\pi}{16} \approx 0.3976$, радиус описанной окружности $R = \frac{r}{\cos \frac{\pi}{16}} \approx 1.0195$. Соответственно, для $\mathrm{PF}_{\text{Форма}}$: начало правой прямоугольной системы координат – т. O_1 (0;0) находится в центре симметрии правильного 16-угольника, соответствующий репер – R_1 (O_1,$\overrightarrow{e_1}$,$\overrightarrow{e_2}$); вектора $\overrightarrow{e_1}$ и $\overrightarrow{e_2}$ являются ортонормированными (с еденичными длинами). Координаты вершин $\mathrm{PF}_{\text{Форма}}$: (≈0.2;1), (≈0.56;≈0.84), (≈0.84;≈0.56), (1;≈0.2), (1;≈-0.2), (≈0.84;≈-0.56), (≈0.56;≈-0.84), (≈0.2;-1), (≈-0.2;-1), (≈-0.56; ≈-0.84), (≈-0.84;≈-0.56), (-1;≈-0.2), (-1;≈0.2), (≈-0.84;≈0.56), (≈-0.56;≈0.84), (≈-0.2;1). Северная сторона $\mathrm{PF}_{\text{Форма}}$, до и после выполнения раундовых про-

цедур, определяется в репере R_1 координатами двух вершин: (≈0.2;1), (≈-0.2;1). Ориентация остальных сторон определяется обходом вокруг часовой стрелки.

Вышеописанная конфигурация продемонстрирована на рис.1.10.

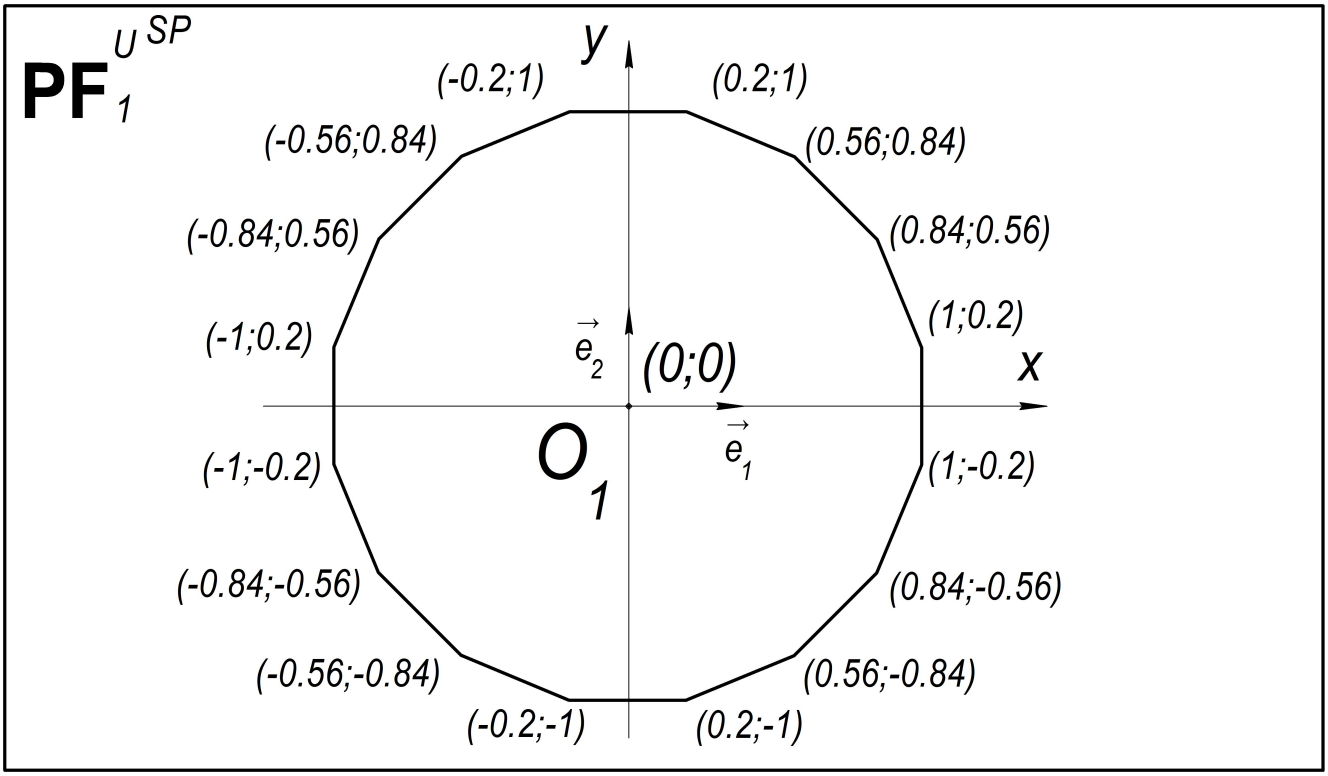


Рисунок 1.10 — Положение правильного 16-угольника на E_2

1.2.6 Логическая взаимосвязь формы, матрицы состояний и блока

Обобщая результаты п.1.2.1-1.2.5, осуществляется возможным получить комплексную (интегрированную) логическую взаимосвязь формы, матрицы состояний и блока – рис.1.11. Взаимосвязь выражается следующим логическим взаимнооднозначным соответствием:

$$l \leftrightarrow \Lambda_{16} \leftrightarrow a_{ij} \leftrightarrow \{b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0\} \leftrightarrow PF_k \qquad (1.3)$$

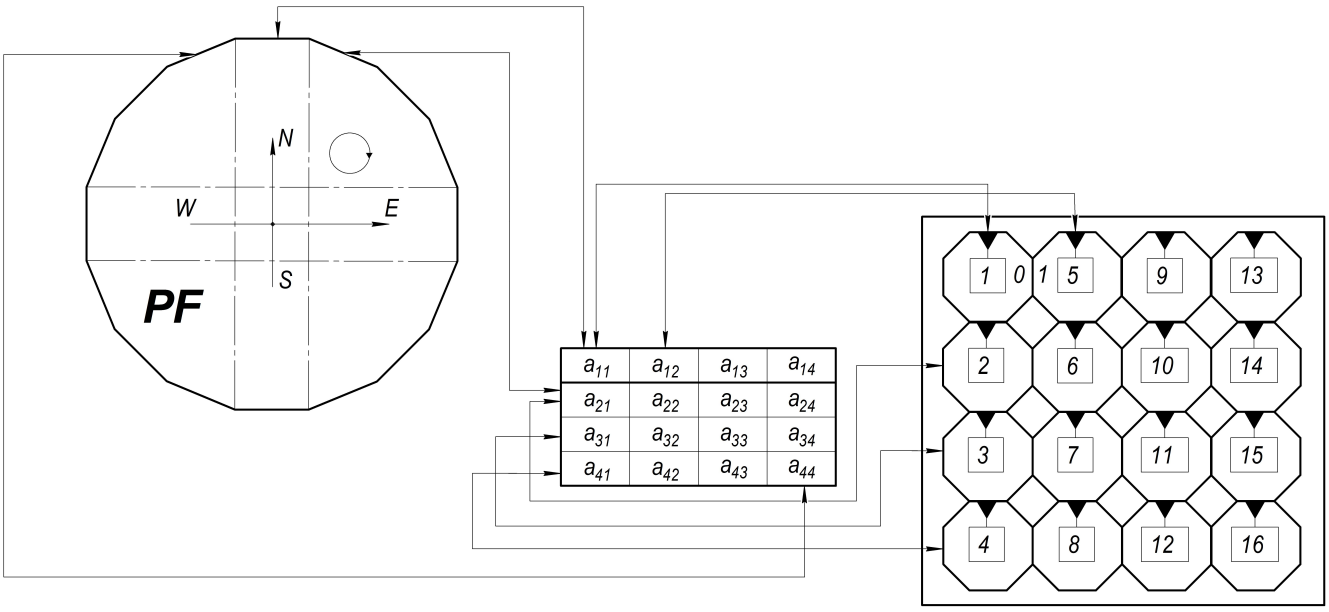


Рисунок 1.11 — Логическая взаимосвязь формы, матрицы состояний и блока

Матрица состояний отображает состояние блока и формы перед, в ходе, и после выполнения всех раундовых процедур.

Продемонстрируем на примере a_{11}. Первому байту a_{11} соответствуют записи в шестнадцатеричной системе счисления – Λ_{16} и двоичной системе счисления – $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$. С $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ образуется PF_1 (п.1.2.1), Λ_{16} соответствует северная сторона $\mathrm{PF}_{\text{Форма}}$ (п.1.2.5). После выполнения раундовых операций над блоком $\mathrm{PF}^{doc}_{S^{16}}$ (PerBits, InvPerBits, ShiftBytes, InvShiftBytes) с PF_1, согласно ориентации сторон (п.1.2.1-1.2.3), снимается запись $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$ и переводится в Λ_{16} и т.д. Аналогично в обратном порядке. *Безусловно, зашифрованный (расшифрованный) блок снимается с матрицы состояний.*

1.2.7 Операция SubBytes

Раундовая операция SubBytes осуществляет подстановку байтов в матрице состояний с помощью таблицы подстановок – рис.1.12. Например, байт $\{52\}$ заменится на $\{00\}$.

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Рисунок 1.12 — Таблица подстановок операции SubBytes

1.2.8 Операция PerBits

Раундовая операция PerBits осуществляет перестановку битов в байте. В контексте данного алгоритма, операция рассматривается как поворот против часовой стрелки отдельно взятого absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} [4, с 589] из комплекса $\mathrm{PF}_{S^{16}}^{doc}$, вокруг центра симметрии на угол $\varphi = \frac{360° \cdot n}{8} = 45° \cdot n, n \in N$, n – количество поворотов. Согласно п.1.2.3, раундовые операции PerBits будем производить над правильными 8-угольниками, образующие каждый из plafales (плафалов) комплекса $\mathrm{PF}_{S^{16}}^{doc}$. Поворот, на данный угол, переводит правильный 8-угольник сам в себя. Множество углов поворотов: $M = \{45°; 90°; 135°; 180°; 225°; 270°; 315°; 360°\}$. Очевидно, что $M \cong Z_8$, Z_8 – кольцо вычетов по модулю 8. Отдельно взятый правильный 8-угольник – PF_k имеет свое собственное количество поворотов, которое определяется формулой:

$$n = \begin{cases} n \equiv f(k) \pmod 8,\ 8 \nmid f(k) \\ 1,\ 8 \mid f(k) \end{cases}$$

$f(k)$ – функция количества поворотов, для каждого раунда имеет индивиду-

альный вид; k – позиция PF_k в сотовой структуре блока (п.1.2.2).

Функция работы поворота $\omega(t)_{PF_d^{uniq}}$ [4, с 590] (в контексте алгоритма: изменение координат вершин правильного 8-угольника в репере R_k,$k = \overline{1;16}$):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.4}$$

$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$ – матрица поворота против часовой стрелки; $(x';y')$ – координаты точки, полученные вращением точки $(x;y)$.

Для всех раундов (1-14) будут составлены таблицы, отображающие вышеописанные функциональные характеристики.

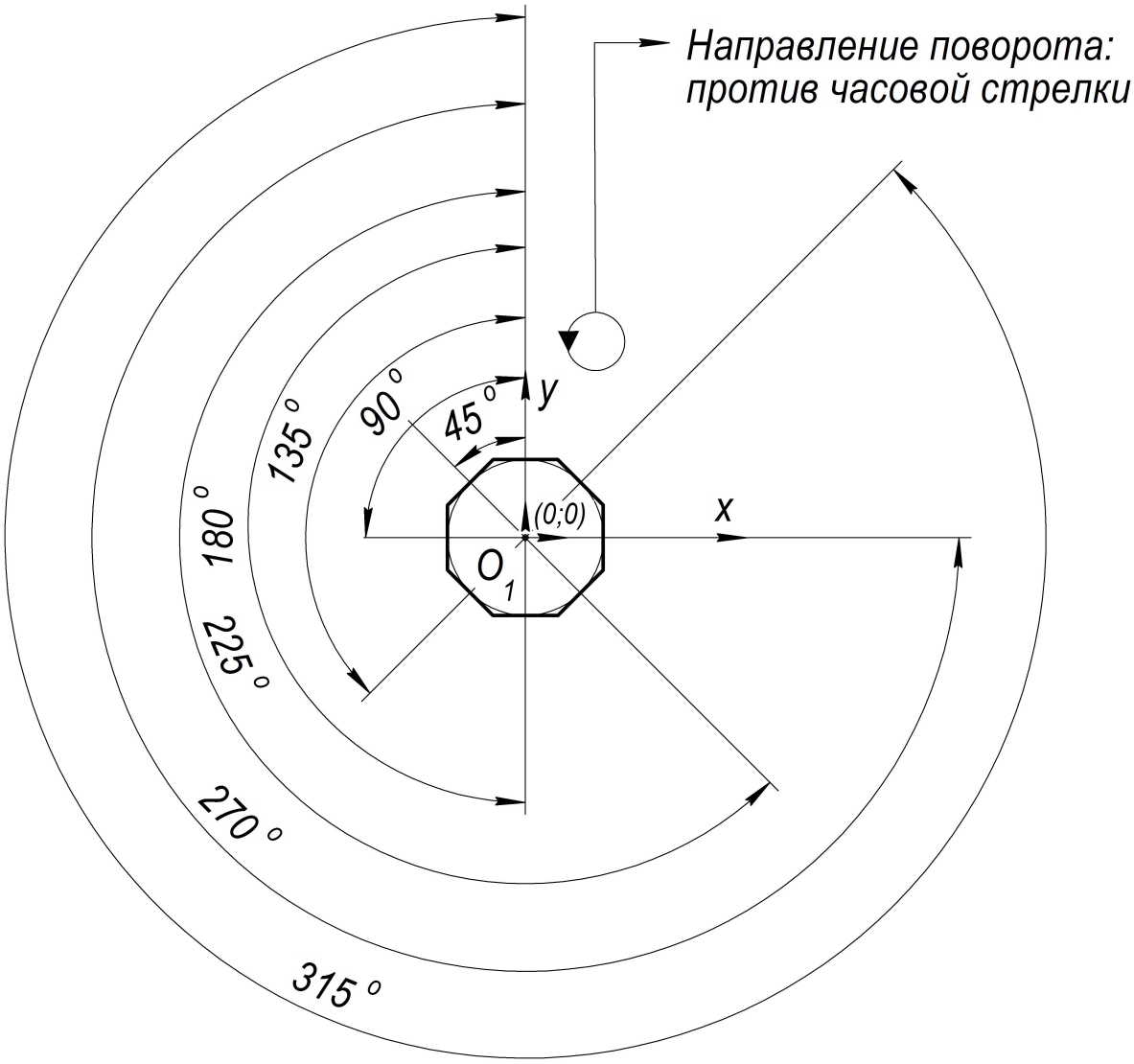


Рисунок 1.13 — Множество углов поворотов для PF_1 комплекса $\mathrm{PF}_{S^{16}}^{doc}$

Таблица 1.3 — Вид матрицы поворота от количества поворотов n

n	φ	$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$
1	$45°$	$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$
2	$90°$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
3	$135°$	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$
4	$180°$	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
5	$225°$	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$
6	$270°$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
7	$315°$	$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$

Раунд 1

$$f(k) = k :$$

Таблица 1.4

PF_k	$f(k)$	n
PF_1	1	1
PF_2	2	2
PF_3	3	3
PF_4	4	4
PF_5	5	5
PF_6	6	6
PF_7	7	7

PF_8	8	1
PF_9	9	1
PF_{10}	10	2
PF_{11}	11	3
PF_{12}	12	4
PF_{13}	13	5
PF_{14}	14	6
PF_{15}	15	7
PF_{16}	16	1

Раунд 2

$$f(k) = k^2 :$$

Таблица 1.5

PF_k	$f(k)$	n
PF_1	1	1
PF_2	4	4
PF_3	9	1
PF_4	16	1
PF_5	25	1
PF_6	36	4
PF_7	49	1
PF_8	64	1
PF_9	81	1
PF_{10}	100	4
PF_{11}	121	1
PF_{12}	144	1
PF_{13}	169	1
PF_{14}	196	4
PF_{15}	225	1

PF_{16}	256	1

Раунд 3

$$f(k) = k^3 :$$

Таблица 1.6

PF_k	$f(k)$	n
PF_1	1	1
PF_2	8	1
PF_3	27	3
PF_4	64	1
PF_5	125	5
PF_6	216	1
PF_7	343	7
PF_8	512	1
PF_9	729	1
PF_{10}	1000	1
PF_{11}	1331	3
PF_{12}	1728	1
PF_{13}	2197	5
PF_{14}	2744	1
PF_{15}	3375	7
PF_{16}	4096	1

Раунд 4

$$f(k) = k^2 + k :$$

Таблица 1.7

PF_k	$f(k)$	n

PF_1	2	2
PF_2	6	6
PF_3	12	4
PF_4	20	4
PF_5	30	6
PF_6	42	2
PF_7	56	1
PF_8	72	1
PF_9	90	2
PF_{10}	110	6
PF_{11}	132	4
PF_{12}	156	4
PF_{13}	182	6
PF_{14}	210	2
PF_{15}	240	1
PF_{16}	272	1

Раунд 5

$$f(k) = k^3 + k :$$

Таблица 1.8

PF_k	$f(k)$	n
PF_1	2	2
PF_2	10	2
PF_3	30	6
PF_4	68	4
PF_5	130	2
PF_6	222	6
PF_7	350	6
PF_8	520	1

PF_9	738	2
PF_{10}	1010	2
PF_{11}	1342	6
PF_{12}	1740	4
PF_{13}	2210	2
PF_{14}	2758	6
PF_{15}	3390	6
PF_{16}	4112	1

Раунд 6

$$f(k) = [\pi^k] :$$

Таблица 1.9

PF_k	$f(k)$	n
PF_1	3	3
PF_2	9	1
PF_3	31	7
PF_4	97	1
PF_5	306	2
PF_6	961	1
PF_7	3020	4
PF_8	9488	1
PF_9	29809	1
PF_{10}	93648	1
PF_{11}	294204	4
PF_{12}	924269	5
PF_{13}	2903677	5
PF_{14}	9122171	3
PF_{15}	28658145	1
PF_{16}	90032220	4

Раунд 7

$$f(k) = 2^k + 1 :$$

Таблица 1.10

PF_k	$f(k)$	n
PF_1	3	3
PF_2	5	5
PF_3	9	1
PF_4	17	1
PF_5	33	1
PF_6	65	1
PF_7	129	1
PF_8	257	1
PF_9	513	1
PF_{10}	1025	1
PF_{11}	2049	1
PF_{12}	4097	1
PF_{13}	8193	1
PF_{14}	16385	1
PF_{15}	32769	1
PF_{16}	65537	1

Раунд 8

$$f(k) = k^3 :$$

Таблица 1.11

PF_k	$f(k)$	n
PF_1	1	1
PF_2	8	1

PF_3	27	3
PF_4	64	1
PF_5	125	5
PF_6	216	1
PF_7	343	7
PF_8	512	1
PF_9	729	1
PF_{10}	1000	1
PF_{11}	1331	3
PF_{12}	1728	1
PF_{13}	2197	5
PF_{14}	2744	1
PF_{15}	3375	7
PF_{16}	4096	1

Раунд 9

$$f(k) = k^2 :$$

Таблица 1.12

PF_k	$f(k)$	n
PF_1	1	1
PF_2	4	4
PF_3	9	1
PF_4	16	1
PF_5	25	1
PF_6	36	4
PF_7	49	1
PF_8	64	1
PF_9	81	1
PF_{10}	100	4

PF_{11}	121	1
PF_{12}	144	1
PF_{13}	169	1
PF_{14}	196	4
PF_{15}	225	1
PF_{16}	256	1

Раунд 10

$$f(k) = 2^k + 1 :$$

Таблица 1.13

PF_k	$f(k)$	n
PF_1	3	3
PF_2	5	5
PF_3	9	1
PF_4	17	1
PF_5	33	1
PF_6	65	1
PF_7	129	1
PF_8	257	1
PF_9	513	1
PF_{10}	1025	1
PF_{11}	2049	1
PF_{12}	4097	1
PF_{13}	8193	1
PF_{14}	16385	1
PF_{15}	32769	1
PF_{16}	65537	1

Раунд 11

$$f(k) = k :$$

Таблица 1.14

PF_k	$f(k)$	n
PF_1	1	1
PF_2	2	2
PF_3	3	3
PF_4	4	4
PF_5	5	5
PF_6	6	6
PF_7	7	7
PF_8	8	1
PF_9	9	1
PF_{10}	10	2
PF_{11}	11	3
PF_{12}	12	4
PF_{13}	13	5
PF_{14}	14	6
PF_{15}	15	7
PF_{16}	16	1

Раунд 12

$$f(k) = k^3 + k :$$

Таблица 1.15

PF_k	$f(k)$	n
PF_1	2	2
PF_2	10	2

PF_3	30	6
PF_4	68	4
PF_5	130	2
PF_6	222	6
PF_7	350	6
PF_8	520	1
PF_9	738	2
PF_{10}	1010	2
PF_{11}	1342	6
PF_{12}	1740	4
PF_{13}	2210	2
PF_{14}	2758	6
PF_{15}	3390	6
PF_{16}	4112	1

Раунд 13

$$f(k) = [\pi^k] :$$

Таблица 1.16

PF_k	$f(k)$	n
PF_1	3	3
PF_2	9	1
PF_3	31	7
PF_4	97	1
PF_5	306	2
PF_6	961	1
PF_7	3020	4
PF_8	9488	1
PF_9	29809	1
PF_{10}	93648	1

PF_{11}	294204	4
PF_{12}	924269	5
PF_{13}	2903677	5
PF_{14}	9122171	3
PF_{15}	28658145	1
PF_{16}	90032220	4

Раунд 14

$$f(k) = k^2 + k :$$

Таблица 1.17

PF_k	$f(k)$	n
PF_1	2	2
PF_2	6	6
PF_3	12	4
PF_4	20	4
PF_5	30	6
PF_6	42	2
PF_7	56	1
PF_8	72	1
PF_9	90	2
PF_{10}	110	6
PF_{11}	132	4
PF_{12}	156	4
PF_{13}	182	6
PF_{14}	210	2
PF_{15}	240	1
PF_{16}	272	1

1.2.9 Операция ShiftBytes

Раундовая операция ShiftBytes осуществляет циклический сдвиг байт в матрице состояний на различные величины. В контексте данного алгоритма рассматривается как параллельный перенос PF_i (п.1.2.2) на PF_j (п.1.2.2); $i \neq j$ комплекса $\mathrm{PF}_{S^{16}}^{doc}$. Параллельный перенос – „lucidification“ of the system of plafales [4, с 592] (осветление сотовой структуры блока) – $\mathrm{LUC}_{PF_{S^{16}}}$.

$$LUC_{PF_{S^{16}}} = \begin{cases} PF_2 \to PF_{14}, PF_{14} \to PF_{10}, PF_{10} \to PF_6, PF_6 \to PF_2 \\ PF_3 \to PF_{11}, PF_{15} \to PF_7, PF_{11} \to PF_3, PF_7 \to PF_{15} \\ PF_4 \to PF_8, PF_{16} \to PF_4, PF_{12} \to PF_{16}, PF_8 \to PF_{12} \end{cases}$$

Согласно п.1.2.3, раундовые операции ShiftBytes будем производить над правильными 8-угольниками, образующие каждый из plafales (плафалов) комплекса $\mathrm{PF}_{S^{16}}^{doc}$. То есть, правильный 8-угольник который занимает 2 позицию, параллельным переносом переходит на 14 позицию и т.д. Фактически, правильный 8-угольник из репера R_i переходит в репер R_j. Вектором указанного перехода выступает $\overrightarrow{O_iO_j}$. Обобщая вышесказанное:

$$\begin{cases} R_2 \to R_{14}, \overrightarrow{O_2O_{14}} = (6,0) \\ R_{14} \to R_{10}, \overrightarrow{O_{14}O_{10}} = (-2,0) \\ R_{10} \to R_6, \overrightarrow{O_{10}O_6} = (-2,0) \\ R_6 \to R_2, \overrightarrow{O_6O_2} = (-2,0) \\ R_3 \to R_{11}, \overrightarrow{O_3O_{11}} = (4,0) \\ R_{15} \to R_7, \overrightarrow{O_{15}O_7} = (-4,0) \\ R_{11} \to R_3, \overrightarrow{O_{11}O_3} = (-4,0) \\ R_7 \to R_{15}, \overrightarrow{O_7O_{15}} = (4,0) \\ R_4 \to R_8, \overrightarrow{O_4O_8} = (2,0) \\ R_{16} \to R_4, \overrightarrow{O_{16}O_4} = (-6,0) \\ R_{12} \to R_{16}, \overrightarrow{O_{12}O_{16}} = (2,0) \\ R_8 \to R_{12}, \overrightarrow{O_8O_{12}} = (2,0) \end{cases}$$

Данная конфигурация выполняется во всех раундах 1-14.

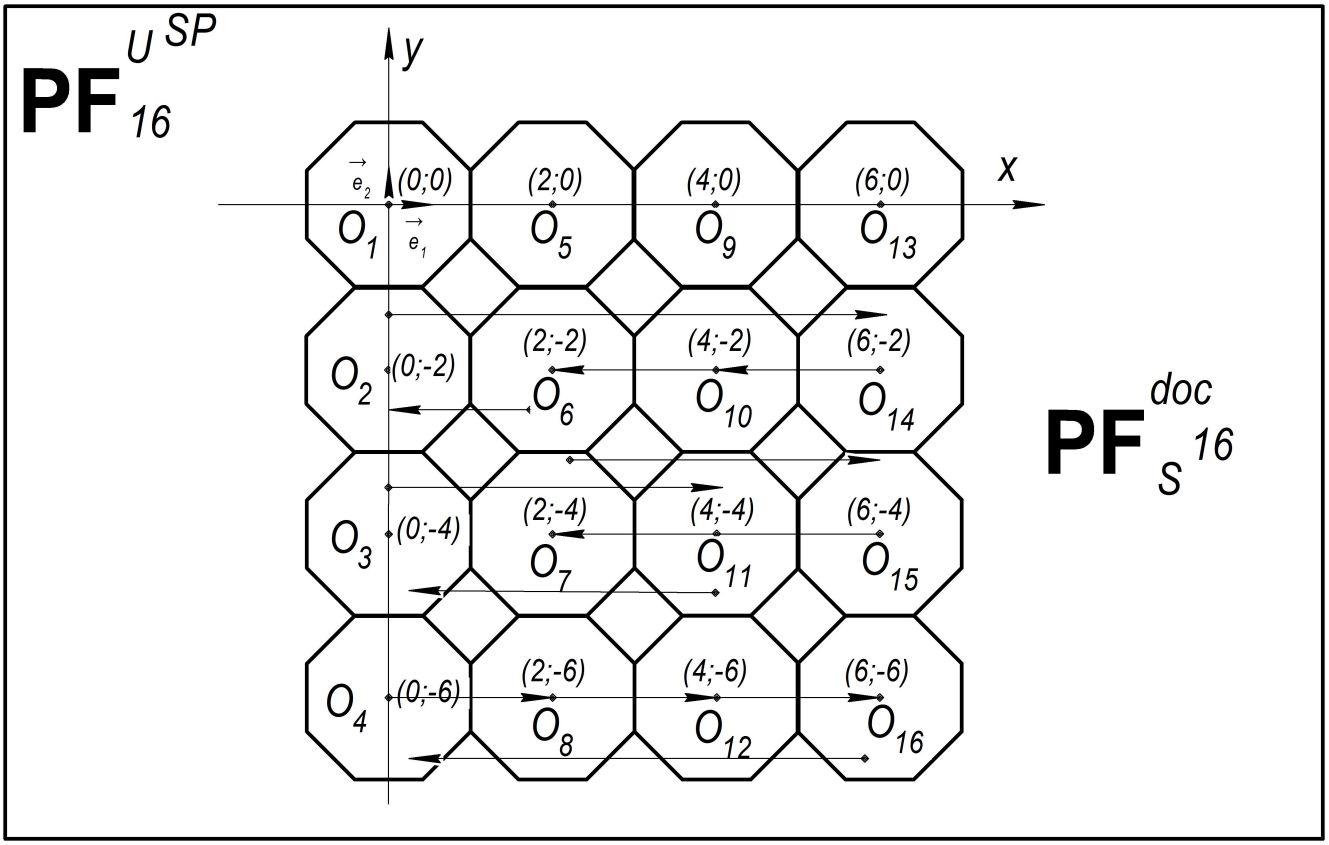


Рисунок 1.14 — Раундовая операция ShiftBytes

1.2.10 Операция PerBytes

Операция PerBytes осуществляет перестановку байт в матрице состояний. В контексте данного алгоритма рассматривается как поворот против часовой стрелки absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} вокруг центра симметрии на угол $\varphi = \frac{360^\circ \cdot n}{16} = 22.5^\circ \cdot n$, n – количество поворотов. Согласно п.1.2.3 и п.1.2.5, раундовые операции PerBytes будем производить над правильным 16-угольником, образующий форму. Поворот, на данный угол, переводит правильный 16-угольник сам в себя. Множество углов поворотов:

$$M = \{22.5^\circ; 45^\circ; 67.5^\circ; 90^\circ; 112.5^\circ; 135^\circ; 157.5^\circ; 180^\circ; 202.5^\circ; 225^\circ; 247.5^\circ; 270^\circ;$$

$$292.5^\circ; 315^\circ; 337.5^\circ; 360^\circ\}$$

Очевидно, что $M \cong Z_{16}$, Z_{16} – кольцо вычетов по модулю 16. Правильный 16-угольник – $\mathrm{PF}_{\text{Форма}}$, в каждом из 13 раундов, имеет свое собственное количество

поворотов, которое определяется формулой:

$$n = \begin{cases} n \equiv f(k) \pmod{16},\ 16 \nmid f(k) \\ 1,\ 16 \mid f(k) \end{cases}$$

$f(k)$ – функция количества поворотов, $f(k) = \frac{(\frac{1+\sqrt{5}}{2})^k - (\frac{1-\sqrt{5}}{2})^k}{\sqrt{5}}$, k – номер раунда.

Функция работы поворота $\omega(t)_{PF_d^{uniq}}$ [4, с 590] (в контексте алгоритма: изменение координат вершин правильного 16-угольника в репере R_1):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.5}$$

$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$ – матрица поворота против часовой стрелки; $(x'; y')$ – координаты точки, полученные вращением точки $(x; y)$.

Для всех раундов (1-13) будут составлены таблицы, отображающие вышеописанные функциональные характеристики.

Таблица 1.18 — Вид матрицы поворота от количества поворотов n

n	φ	$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$
1	$22.5°$	$\begin{bmatrix} 0.92 & -0.38 \\ 0.38 & 0.92 \end{bmatrix}$
2	$45°$	$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$
3	$67.5°$	$\begin{bmatrix} 0.38 & -0.92 \\ 0.92 & 0.38 \end{bmatrix}$
4	$90°$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
5	$112.5°$	$\begin{bmatrix} -0.38 & -0.92 \\ 0.92 & -0.38 \end{bmatrix}$
6	$135°$	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$

7	157.5°	$\begin{bmatrix} -0.92 & -0.38 \\ 0.38 & -0.92 \end{bmatrix}$
8	180°	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
9	202.5°	$\begin{bmatrix} -0.92 & 0.38 \\ -0.38 & -0.92 \end{bmatrix}$
10	225°	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$
11	247.5°	$\begin{bmatrix} -0.38 & 0.92 \\ -0.92 & -0.38 \end{bmatrix}$
12	270°	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
13	292.5°	$\begin{bmatrix} 0.38 & 0.92 \\ -0.92 & 0.38 \end{bmatrix}$
14	315°	$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$
15	337.5°	$\begin{bmatrix} 0.92 & 0.38 \\ -0.38 & 0.92 \end{bmatrix}$

Раунд 1

Таблица 1.19

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	1	1

Раунд 2

Таблица 1.20

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	1	1

Раунд 3

Таблица 1.21

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	2	2

Раунд 4

Таблица 1.22

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	3	3

Раунд 5

Таблица 1.23

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	5	5

Раунд 6

Таблица 1.24

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	8	8

Раунд 7

Таблица 1.25

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	13	13

Раунд 8

Таблица 1.26

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	21	5

Раунд 9

Таблица 1.27

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	34	2

Раунд 10

Таблица 1.28

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	55	7

Раунд 11

Таблица 1.29

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	89	9

Раунд 12

Таблица 1.30

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	144	1

Раунд 13

Таблица 1.31

$\text{PF}_{\text{Форма}}$	$f(k)$	n
$\text{PF}_{\text{Форма}}$	233	9

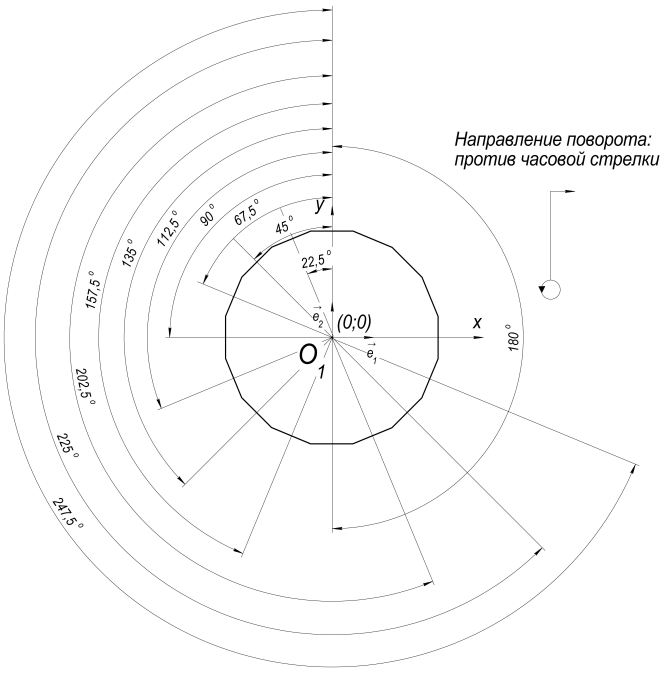


Рисунок 1.15 — Множество углов поворотов для $PF_{Форма}$

1.2.11 InvSubBytes

Это преобразование обратно преобразованию SubBytes. Раундовая операция InvSubBytes осуществляет подстановку байтов в матрице состояний с помощью обратной таблицы подстановок – рис.1.16.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Рисунок 1.16 — Таблица подстановок операции InvSubBytes

1.2.12 InvPerBits

Это преобразование обратно преобразованию PerBits. Раундовая операция InvPerBits осуществляет перестановку битов в байте. В контексте данного алгоритма, операция рассматривается как поворот вокруг часовой стрелки отдельно взятого absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} [4, с 589] из комплекса $PF_{S^{16}}^{doc}$, вокруг центра симметрии на угол $\varphi = \frac{360° \cdot n}{8} = 45° \cdot n, n \in N$, n – количество поворотов. Согласно п.1.2.3, раундовые операции InvPerBits будем производить над правильными 8-угольниками, образующие каждый из plafales (плафалов) комплекса $PF_{S^{16}}^{doc}$. Поворот, на данный угол, переводит правильный 8-угольник сам в себя. Множество углов поворотов: $M = \{45°; 90°; 135°; 180°; 225°; 270°; 315°; 360°\}$. Очевидно, что $M \cong Z_8$, Z_8 – кольцо вычетов по модулю 8. Отдельно взятый правильный 8-угольник – PF_k имеет свое собственное количество поворотов, которое определяется формулой:

$$n = \begin{cases} n \equiv f(k) \ \ (mod\, 8),\ 8 \nmid f(k) \\ 1,\ 8 \mid f(k) \end{cases}$$

$f(k)$ – функция количества поворотов, для каждого раунда имеет индивидуальный вид; k – позиция PF_k в сотовой структуре блока (п.1.2.2).

Функция работы поворота $\omega(t)_{PF_d^{uniq}}$ [4, с 590] (в контексте алгоритма: изменение координат вершин правильного 8-угольника в репере R_k,$k = \overline{1;16}$):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.6}$$

$\begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix}$ – матрица поворота вокруг часовой стрелки; $(x'; y')$ – координаты точки, полученные вращением точки $(x; y)$.

Таблица 1.32 — Вид матрицы поворота от количества поворотов n

n	φ	$\begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix}$
1	45°	$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$

2	$90°$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
3	$135°$	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$
4	$180°$	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
5	$225°$	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$
6	$270°$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
7	$315°$	$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$

Раунды 1-14 выполняются в порядке 14-1 раундовой операции PerBits. То есть, 1-ый раунд операции InvPerBits – 14 раунд операции PerBits и т.д.

1.2.13 InvShiftBytes

Это преобразование обратно преобразованию ShiftBytes. Раундовая операция InvShiftBytes осуществляет циклический сдвиг байт в матрице состояний на различные величины. В контексте данного алгоритма рассматривается как параллельный перенос PF_j (п.1.2.2) на PF_i (п.1.2.2); $j \neq i$ комплекса $\mathrm{PF}^{doc}_{S^{16}}$. Параллельный перенос – „lucidification“ of the system of plafales [4, с 592] (осветление сотовой структуры блока) – $\mathrm{LUC}_{PF_{S^{16}}}$.

$$LUC_{PF_{S^{16}}} = \begin{cases} PF_{14} \to PF_2, PF_{10} \to PF_{14}, PF_6 \to PF_{10}, PF_2 \to PF_6 \\ PF_{11} \to PF_3, PF_7 \to PF_{15}, PF_3 \to PF_{11}, PF_{15} \to PF_7 \\ PF_8 \to PF_4, PF_4 \to PF_{16}, PF_{16} \to PF_{12}, PF_{12} \to PF_8 \end{cases}$$

Согласно п.1.2.3, раундовые операции InvShiftBytes будем производить над правильными 8-угольниками, образующие каждый из plafales (плафалов) комплекса $\mathrm{PF}^{doc}_{S^{16}}$. То есть, правильный 8-угольник который занимает 14 позицию,

параллельным переносом переходит на 2 позицию и т.д. Фактически, правильный 8-угольник из репера R_j переходит в репер R_i. Вектором указанного перехода выступает $\overrightarrow{O_jO_i} = -\overrightarrow{O_iO_j}$. Обобщая вышесказанное:

$$\begin{cases} R_{14} \to R_2, \overrightarrow{O_2O_{14}} = (-6, 0) \\ R_{10} \to R_{14}, \overrightarrow{O_{14}O_{10}} = (2, 0) \\ R_6 \to R_{10}, \overrightarrow{O_{10}O_6} = (2, 0) \\ R_2 \to R_6, \overrightarrow{O_6O_2} = (2, 0) \\ R_{11} \to R_3, \overrightarrow{O_3O_{11}} = (-4, 0) \\ R_7 \to R_{15}, \overrightarrow{O_{15}O_7} = (4, 0) \\ R_3 \to R_{11}, \overrightarrow{O_{11}O_3} = (4, 0) \\ R_{15} \to R_7, \overrightarrow{O_7O_{15}} = (-4, 0) \\ R_8 \to R_4, \overrightarrow{O_4O_8} = (-2, 0) \\ R_4 \to R_{16}, \overrightarrow{O_{16}O_4} = (6, 0) \\ R_{16} \to R_{12}, \overrightarrow{O_{12}O_{16}} = (-2, 0) \\ R_{12} \to R_8, \overrightarrow{O_8O_{12}} = (-2, 0) \end{cases}$$

Данная конфигурация выполняется во всех раундах 1-14.

1.2.14 InvPerBytes

Это преобразование обратно преобразованию PerBytes. Операция InvPerBytes осуществляет перестановку байт в матрице состояний. В контексте данного алгоритма рассматривается как поворот вокруг часовой стрелки absolutely dynamic plafal (абсолютно динамического плафала) – PF_{ad}^{uniq} вокруг центра симметрии на угол $\varphi = \frac{360° \cdot n}{16} = 22.5° \cdot n$, n – количество поворотов. Согласно п.1.2.3 и п.1.2.5, раундовые операции InvPerBytes будем производить над правильным 16-угольником, образующий форму. Поворот, на данный угол, переводит правильный 16-угольник сам в себя. Множество углов поворотов:

$$M = \{22.5°; 45°; 67.5°; 90°; 112.5°; 135°; 157.5°; 180°; 202.5°; 225°; 247.5°; 270°;$$

$$292.5°; 315°; 337.5°; 360°\}$$

Очевидно, что $M \cong Z_{16}$, Z_{16} – кольцо вычетов по модулю 16. Правильный 16-угольник – $PF_{Форма}$, в каждом из 13 раундов, имеет свое собственное количество поворотов, которое определяется формулой:

$$n = \begin{cases} n \equiv f(k) \ \ (mod\, 16),\ 16 \nmid f(k) \\ 1,\ 16 \mid f(k) \end{cases}$$

$f(k)$ – функция количества поворотов, $f(k) = \frac{(\frac{1+\sqrt{5}}{2})^{15-k} - (\frac{1-\sqrt{5}}{2})^{15-k}}{\sqrt{5}}$, k – номер раунда.

Функция работы поворота $\omega(t)_{PF_d^{uniq}}$ [4, с 590] (в контексте алгоритма: изменение координат вершин правильного 16-угольника в репере R_1):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.7}$$

$\begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix}$ – матрица поворота вокруг часовой стрелки; $(x'; y')$ – координаты точки, полученные вращением точки $(x; y)$.

Таблица 1.33 — Вид матрицы поворота от количества поворотов n

n	φ	$\begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix}$
1	$22.5°$	$\begin{bmatrix} 0.92 & 0.38 \\ -0.38 & 0.92 \end{bmatrix}$
2	$45°$	$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$
3	$67.5°$	$\begin{bmatrix} 0.38 & 0.92 \\ -0.92 & 0.38 \end{bmatrix}$
4	$90°$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
5	$112.5°$	$\begin{bmatrix} -0.38 & 0.92 \\ -0.92 & -0.38 \end{bmatrix}$
6	$135°$	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$

7	157.5°	$\begin{bmatrix} -0.92 & 0.38 \\ -0.38 & -0.92 \end{bmatrix}$
8	180°	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
9	202.5°	$\begin{bmatrix} -0.92 & -0.38 \\ 0.38 & -0.92 \end{bmatrix}$
10	225°	$\begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$
11	247.5°	$\begin{bmatrix} -0.38 & -0.92 \\ 0.92 & -0.38 \end{bmatrix}$
12	270°	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
13	292.5°	$\begin{bmatrix} 0.38 & -0.92 \\ 0.92 & 0.38 \end{bmatrix}$
14	315°	$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$
15	337.5°	$\begin{bmatrix} 0.92 & -0.38 \\ 0.38 & 0.92 \end{bmatrix}$

Раунды 2-14 выполняются в порядке 13-1 раундовой операции PerBytes. То есть, 2-ой раунд операции InvPerBytes – 13 раунд операции PerBytes и т.д.

1.3 Алгоритм выработки ключей (Key Schedule)

Раундовые ключи получаются из ключа шифрования посредством алгоритма выработки ключей. Он содержит два компонента: расширение ключа и выбор раундового ключа. Основополагающие принципы алгоритма выглядят следующим образом:

1. Общее число битов раундовых ключей равно длине блока, умноженной на число раундов, плюс 1. То есть, 128·(14+1)=1920 бит = 240 байт. Размерность каждого из пятнадцати раундовых ключей – 16 байт.

2. Ключ шифрования расширяется в расширенный ключ.

3. Раундовые ключи берутся из расширенного ключа следующим образом: первый раундовый ключ содержит первые 16 байт, второй – следующие 16 байт и т.д.

Процедура ExpandKey

• Первые шестнадцать байтов ключа шифрования остаются неизменными.

Ко вторым шестнадцати байтам ключа шифрования применяется процедура SubBytes (рис.1.12). Последующие байты расширенного ключа определяются следующим образом:

$$a_{32+i} = a_{31+i} \oplus a_i, i = \overline{1; 208}$$

Каждый последующий байт a_{32+i} получается посредством XOR предыдущего байта и байта, на 32 позиций ранее.

• К каждой из последовательностей: $\{\overline{a_{33}; a_{48}}\}$; $\{\overline{a_{49}; a_{64}}\}$; $\{\overline{a_{65}; a_{80}}\}$; $\{\overline{a_{81}; a_{96}}\}$; $\{\overline{a_{97}; a_{112}}\}$; $\{\overline{a_{113}; a_{128}}\}$; $\{\overline{a_{129}; a_{144}}\}$; $\{\overline{a_{145}; a_{160}}\}$; $\{\overline{a_{161}; a_{176}}\}$; $\{\overline{a_{177}; a_{192}}\}$; $\{\overline{a_{193}; a_{208}}\}$; $\{\overline{a_{209}; a_{224}}\}$; $\{\overline{a_{225}; a_{240}}\}$ применяются однократно процедуры SubBytes, PerBits, ShiftByte. Раунды (1-13) процедуры PerBits применяются в порядке указанных последовательностей. То есть, к последовательности $\{\overline{a_{33}; a_{48}}\}$ применяется 1-ый раунд PerBits (п.1.2.8), к последовательности $\{\overline{a_{49}; a_{64}}\}$ применяется 2-ой раунд PerBits и т.д. Полученные, таким образом, матрицы состояний (п.1.2.4) и являются, соответственно, раундовыми ключами 3-15.

Процедура AddRoundKey

В данной операции матрица состояний раундового ключа добавляется к матрице состояний блока посредством простого поразрядного XOR (используя правила сложения квадратных матриц). Для процедуры зашифрования, к матрице открытого текста добавляются первые шестнадцать байтов ключа шифрования и т.д. Для процедуры расшифрования, раундовые ключи используются в обратном порядке.

1.4 Имитовставка

Для обеспечения имитозащиты открытых данных, состоящих из M количества 128-разрядных блоков $T_o^{(1)}$, $T_o^{(2)}$,..., $T_o^{(M)}$ где $M > 2$, вырабатывается допол-

нительный блок из 64 бит (имитовставка I_1). Процесс выработки имитовставки осуществляется в режиме шифрования CBC. Количество раундов зашифрования для каждого блока – 7. Из полученного конечного значения выбирается отрезок I_1 (имитовставка) длиной 64 бита. Поступившие зашифрованные данные расшифровываются, из полученных блоков открытых данных, аналогично описанному выше, вырабатывается имитовставка I_2, которая затем сравнивается с имитовставкой I_1 полученной вместе с зашифрованными данными. В случае несовпадения имитовставок полученные данные считаются ложными. Выработка имитовставки может производится или перед зашифрованием (после расшифрования) всего сообщения, или параллельно с зашифрованием (расшифрованием) по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (адресную часть, отметку времени, синхропосылку и т.д.) и не зашифровываются. Вероятность навязывания ложных данных равна 2^{-64}.

1.5 Механизм распределения ключей

В авторизированной системе из n пользователей, механизм распределения ключей выглядит следующим образом – рис.1.18:

• В системе существует следующая иерархия ключей:

•• Мастер ключ K_{Master}. Учитывая главенствующую роль в иерархии мастер-ключа, используемого в течении длительного времени, его защите уделяется особое внимание: 1. Мастер-ключ хранится в защищенном от считывания, записи и разрушающих воздействий модуле системы защиты. 2. Мастер-ключ распространяется неэлектронным способом, исключающим его компрометацию. 3. В системе существует способ проверки аутентичности мастер-ключа – рис.1.17.

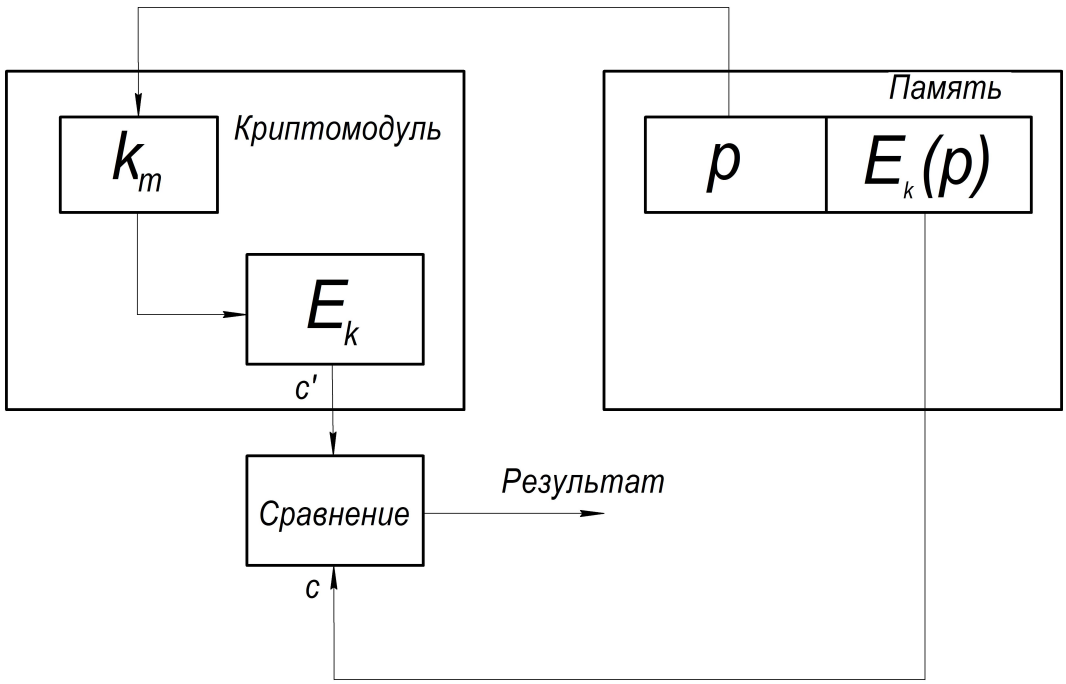


Рисунок 1.17 — Схема аутентификации мастер-ключа

•• K_1, K_2 – Долговременные ключи шифрования данных и ключей.

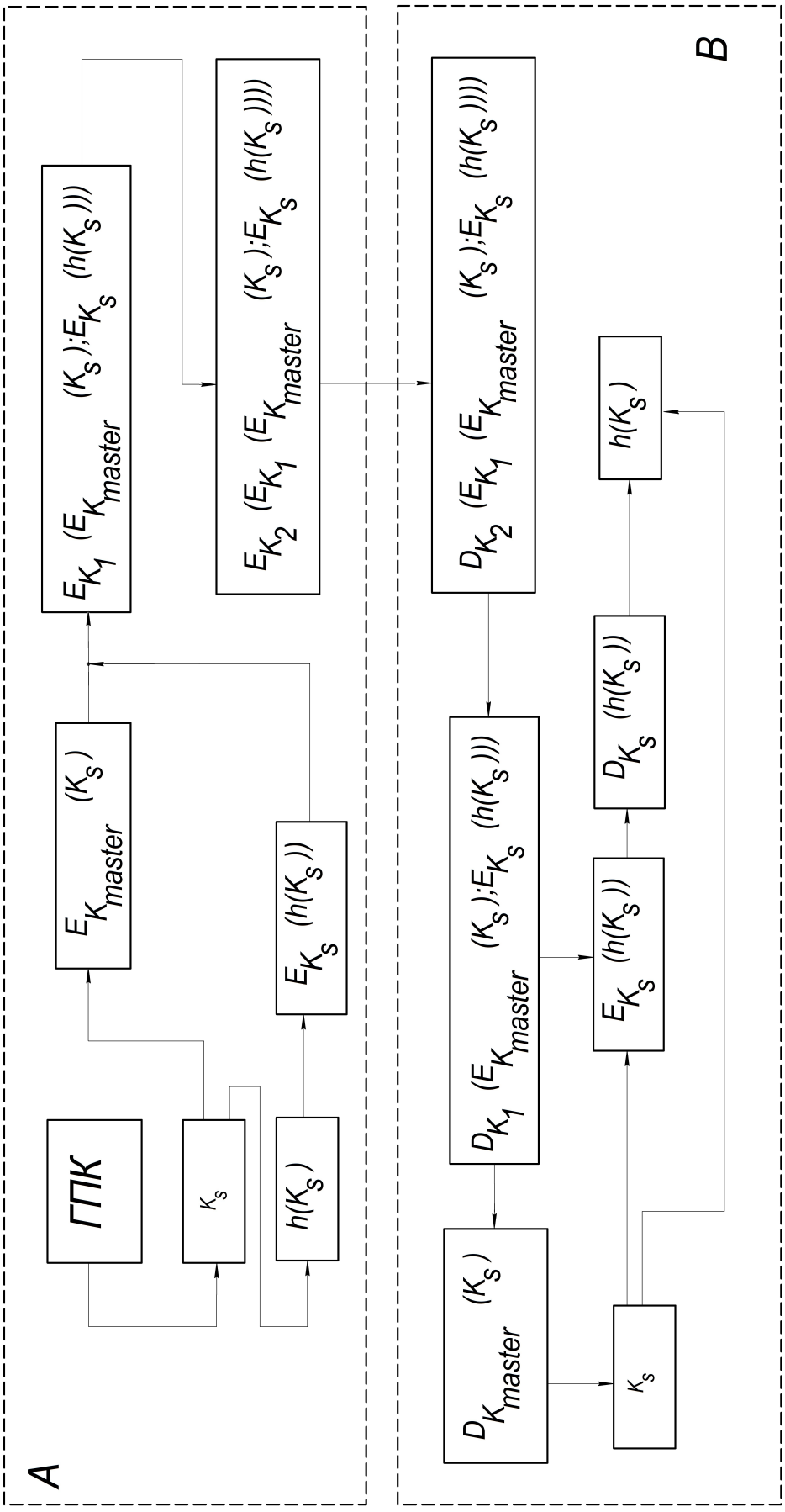


Рисунок 1.18 — Механизм распределения ключей

Отправитель A, снимает с ГПК двоичную 256-разрядную последовательность (ключ распределения – K_S), находит хеш данного ключа – $h(K_S)$, зашифровывает $E_{K_S}(h(K_S))$: хеш $h(K_S)$ на ключе K_S, зашифровывает $E_{K_{Master}}(K_S)$: ключ K_S на ключе K_{Master}; зашифровывает $E_{K_1}(E_{K_{Master}}(K_S); E_{K_S}(h(K_S)))$ на ключе K_1; зашифровывает $E_{K_2}(E_{K_1}(E_{K_{Master}}(K_S); E_{K_S}(h(K_S))))$ на ключе K_2.

Получатель B, получает $E_{K_2}(E_{K_1}(E_{K_{Master}}(K_S); E_{K_S}(h(K_S))))$ и расшифровывает ее – $D_{K_2}(E_{K_1}(E_{K_{Master}}(K_S); E_{K_S}(h(K_S))))$ на ключе K_2; расшифровывает – $D_{K_1}(E_{K_{Master}}(K_S); E_{K_S}(h(K_S)))$ на ключе K_1, в результате получаются зашифрованные последовательности $E_{K_S}(h(K_S))$ и $E_{K_{Master}}(K_S)$. В дальнейшем, получатель расшифровывает – $D_{K_{Master}}(K_S)$ на ключе K_{Master} и получает ключ распределения – K_S; расшифровывает – $D_{K_S}(h(K_S))$ на ключе K_S, в результате получает хеш $h(K_S)$ и сравнивает его с хешем, полученным на ключе K_S.

ЛИТЕРАТУРА

1. Брассар Ж. Современная криптология: Пер. с англ. М.: ПОЛИМЕД, 1999.
2. Topchyi D. The theory of plafales: the proof of P versus NP problem / D. Topchyi. - Best Global Publishing, 2011. - 634 p.
3. Topchyi D. The theory of plafales: the proof algorithms for millennium problems / D. Topchyi. - Best Global Publishing, 2013. - 695 p.
4. Topchyi D. The theory of plafales: the proof algorithms for millennium problems / D. Topchyi. - Best Global Publishing, 2013. - 695 p. - Режим доступа к ресурсу: http://eleanor-cms.ru/uploads/book.pdf
5. Topchyi D. The theory of plafales: Applications of new cryptographic algorithms and platforms in Military complex, IT, Banking system, Financial market / D. Topchyi. - XLII KONFERENCJA ZASTOSOWAN MATEMATYKI, 2013. - Режим доступа к ресурсу: http://www.impan.pl/ zakopane/42/Topchyi.pdf
6. XLII KONFERENCJA ZASTOSOWAN MATEMATYKI, 2013. - Режим доступа к ресурсу: http://www.impan.pl/KZM/42/
7. Осмоловский С. А. Стохастические методы передачи данных. М.: Радио и связь, 1991.