Ежегодная международная научно-практическая конференция
«РусКрипто'2020»

# Параметризованная криптографическая хеш-функция Hamsi-$n$.

Ермаков Кирилл Дмитриевич,
Студент НИЯУ МИФИ

# Цель работы

Цель работы — создание параметризованной криптографической хеш-функции, параметром которой является длина результирующего хеш-кода.

Для достижения заданной цели были поставлены следующие задачи:

- Анализ существующей хеш-функции для последующей модификации;
- Получение параметризованной версии выбранной хеш-функции;
- Исследование полученного алгоритма.

# Криптографическая хеш-функция Hamsi

- Расширение сообщения $E: \{0,1\}^m \rightarrow \{0,1\}^n$
- Конкатенация сообщений $C: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^s$
- Нелинейное преобразование $P, P_f: \{0,1\}^s \rightarrow \{0,1\}^s$
- Усечение сообщения $T: \{0,1\}^s \rightarrow \{0,1\}^n$

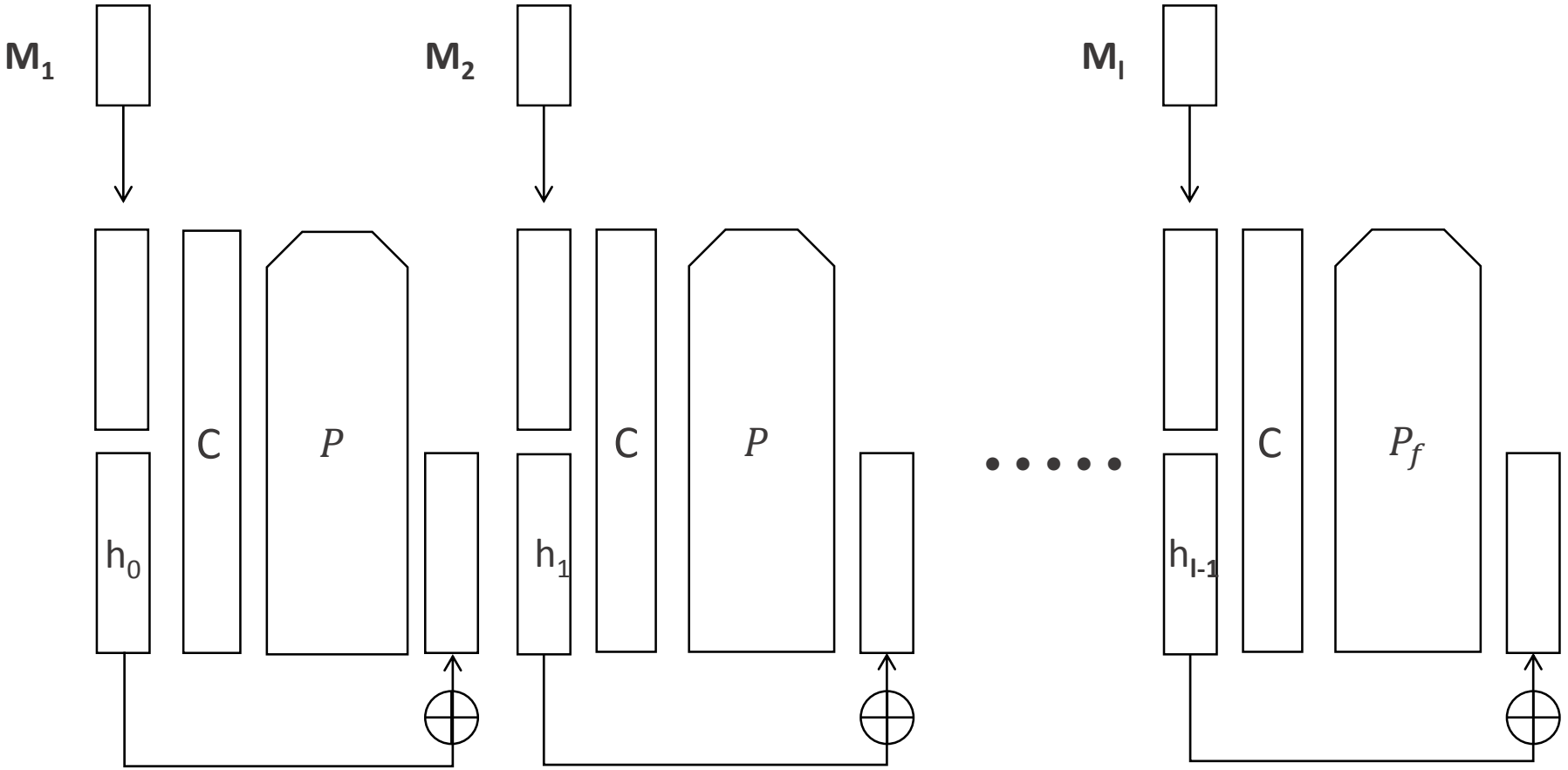
Hamsi-256:

$$h_i = \left(T \circ P \circ C(E(M_i), h_{i-1})\right) \oplus h_{i-1}, h_0 = iv_{256}, 0 < i < l$$
$$h = \left(T \circ P_f \circ C(E(M_l), h_{l-1})\right) \oplus h_{l-1}, m = 32, n = 256, s = 512$$

Hamsi-512:

$$h_i = \left(T \circ P \circ C(E(M_i), h_{i-1})\right) \oplus h_{i-1}, h_0 = iv_{512}, 0 < i < l$$
$$h = \left(T \circ P_f \circ C(E(M_l), h_{l-1})\right) \oplus h_{l-1}, m = 64, n = 512, s = 1024$$

# Схема работы алгоритма




- $M_i$ - блок открытого текста
- $h_i$ - значение хеш-кода на $i$-ой итерации
- C - конкатенация сообщений
- $P$, $P_f$ - нелинейное преобразование

# Анализ линейного преобразования

В нелинейных преобразованиях $\boldsymbol{P}$ и $\boldsymbol{P_f}$ используется следующее линейное преобразование $\boldsymbol{L}$ над векторами вида (a, b, c, d):

$$a := a \lll 13$$
$$c := c \lll 3$$
$$b := b \oplus a \oplus c$$
$$d := d \oplus c \oplus (a \ll 3)$$
$$b := b \lll 1$$

$$d := d \lll 7$$
$$a := a \oplus b \oplus d$$
$$c := c \oplus d \oplus (b \ll 7)$$
$$a := a \lll 5$$
$$c := c \lll 22$$

# Коэффициент рассеивания

Показатель рассеивания (англ. branch number) линейного преобразования $F$ равняется:

$$min_{a \neq 0}\left(W(a) + W\left(F(a)\right)\right)$$

Методом перебора для данного преобразования было найдено, что при векторе вида

$$a = c = d = (0, \dots, 0)$$
$$b = (0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, 0, \dots, 0), x_1, \dots, x_7 \in F_2$$

данная сумма является минимальной и равна **3**.

# Класс линейных преобразований

Вид линейного преобразования:

$$x_1 := x_1 \oplus x_2 \oplus x_3$$
$$x_4 := x_4 \oplus x_2 \oplus (x_3 \ll c_1)$$
$$x_1 := (x_1 \lll c_2)$$
$$x_4 := (x_4 \lll c_3)$$
$$x_2 := x_2 \oplus x_1 \oplus x_4$$
$$x_3 := x_3 \oplus x_1 \oplus (x_4 \ll c_4)$$

где $x_1 \dots x_4 \in F_2^n, c_i \in \{1 \dots n-1\}$

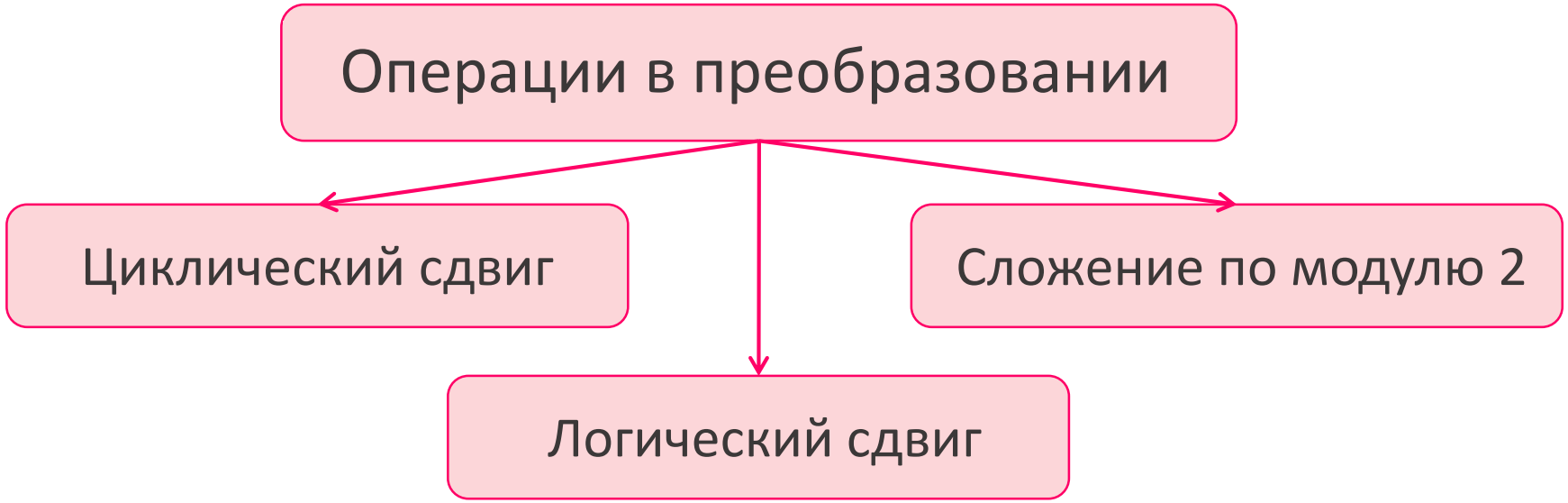
Вектора, при которых значение коэффициента рассеивания минимально, выглядят следующим образом

$x_1 = x_2 = x_4 = (0, \dots, 0), x_3$ зависит от значений битовых сдвигов

$x_1 = x_2 = x_3 = (0, \dots, 0), x_4$ зависит от значений битовых сдвигов

# Инвариантные подпространства

Для полученного класса линейных преобразований был осуществлен поиск инвариантных подпространств для проверки уязвимости данного класса к атакам основанным на инвариантных подпространствах.



# Инвариантные подпространства

Полученное инвариантное подпространство для класса линейных преобразований имеет вид:

$$(a_1 a_1 a_1 \dots d_1 d_1 d_1 a_2 a_2 a_2 \dots d_2 d_2 d_2 a_3 a_3 a_3 \dots d_3 d_3 d_3 a_4 a_4 a_4 \dots d_4 d_4 d_4),$$
$$a_i, \dots, d_i \in F_2$$

Для того, чтобы не было инвариантных подпространств необходимо и достаточно использовать 2 сдвига **a** и **b**, что:

**НОД(a,b)=1**

# Параметризованная хеш-функция Hamsi-$n$

Параметризованную версию алгоритма Hamsi-$n$ можно представить с помощью следующих преобразований:

- Расширение сообщения $E: \{0,1\}^{n/8} \rightarrow \{0,1\}^{n}$
- Конкатенация сообщений $C: \{0,1\}^{n} \times \{0,1\}^{n} \rightarrow \{0,1\}^{2 \cdot n}$
- Нелинейное преобразование $P, P_f: \{0,1\}^{2 \cdot n} \rightarrow \{0,1\}^{2 \cdot n}$
- Усечение сообщения $T: \{0,1\}^{2 \cdot n} \rightarrow \{0,1\}^{n}$

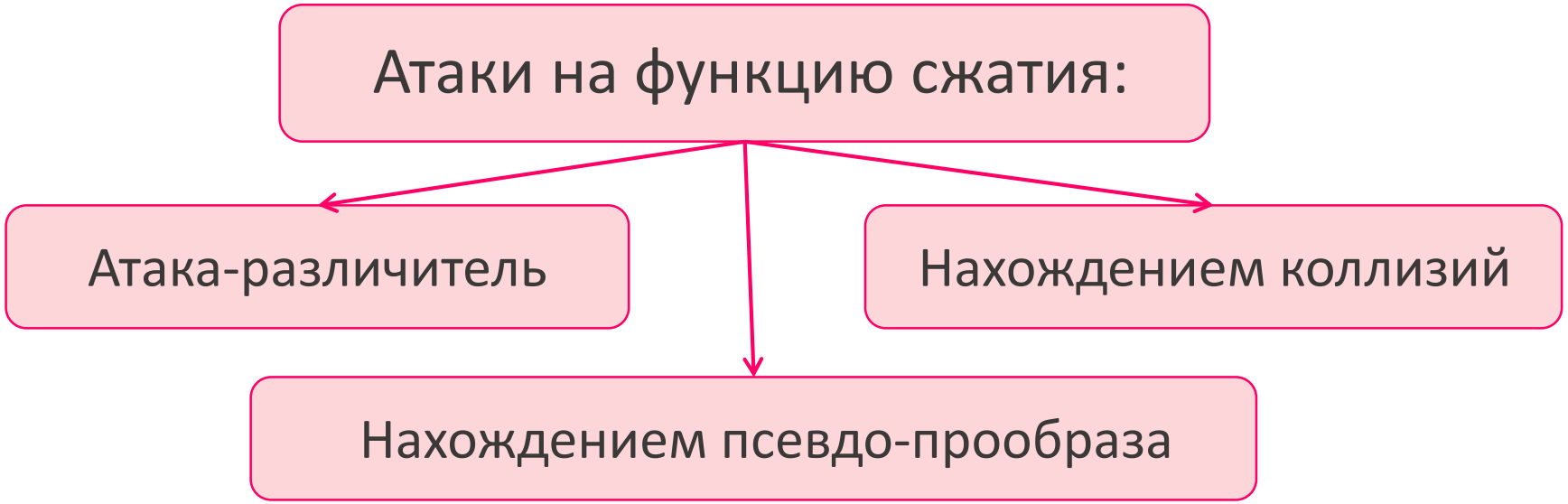
$$h_i = \left(T \circ P \circ C(E(M_i), h_{i-1})\right) \oplus h_{i-1}, h_0 = iv_n, 0 < i < l$$
$$h = \left(T \circ P_f \circ C(E(M_l), h_{l-1})\right) \oplus h_{l-1}$$

# Особенности параметризованной версии

- Размер блоков открытого текста равен $\frac{n}{8}$ бит;
- Внутреннее состояние представлено в виде матрицы 4x4 слова по $\frac{n}{8}$ бит;
- При расширении сообщения могут использоваться линейные коды над полями $F_2, F_4, F_{16}$;
- Используется линейное преобразование из полученного класса л/п.

# Рассмотренные атаки на Hamsi

Основными работами по криптоанализу хеш-функции Hamsi являются атаки на функцию сжатия Hamsi-256. Все данные атаки направлены на функцию сжатия криптографической хеш-функции, не понижая при этом стойкость самой функции.



# Алгоритм поиска неактивных бит

1. Выбрать столбец матрицы внутреннего состояния, сделать начальное внутреннее состояние нулевым;

2. Добавить разницу весом в 1 или 2 бита цепному значению в выбранном столбце;

3. Наблюдать за возможным изменением внутреннего состояния с учетом добавленной разницы на протяжении нескольких раундов.

4. После прохождения всех раундов, неотмеченные биты результата не будут меняться с вероятностью 1 при добавлении выбранной разницы.

# Изменения в параметризованном алгоритме

Проанализировав атаки и алгоритм по поиску неактивных бит, в алгоритм были внесены следующие изменения:

- Изменение S-блока
- Изменение класса линейных преобразований
- Изменение слоя линейных преобразований

# Изменение s-блока

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 4 | 2 |
| **2** | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 4 |
| **3** | 0 | 4 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 2 | 2 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 4 | 4 | 0 | 4 | 0 | 0 |
| **5** | 0 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| **6** | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 |
| **7** | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 |
| **8** | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 |
| ***9*** | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 4 | 2 | 2 | 2 | 2 | 0 | 0 | 0 |
| ***a*** | **0** | **0** | **2** | **0** | **2** | **0** | **4** | **0** | **2** | **0** | **4** | **0** | **0** | **0** | **2** | **0** |
| ***b*** | 0 | 4 | 0 | 0 | 2 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 |
| ***c*** | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 4 | 0 | 4 | 2 | 0 |
| ***d*** | 0 | 4 | 2 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| ***e*** | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 4 | 2 | 0 |
| ***f*** | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 |

# Изменение s-блока

Используемые в блочном шифре Serpent s-блоки

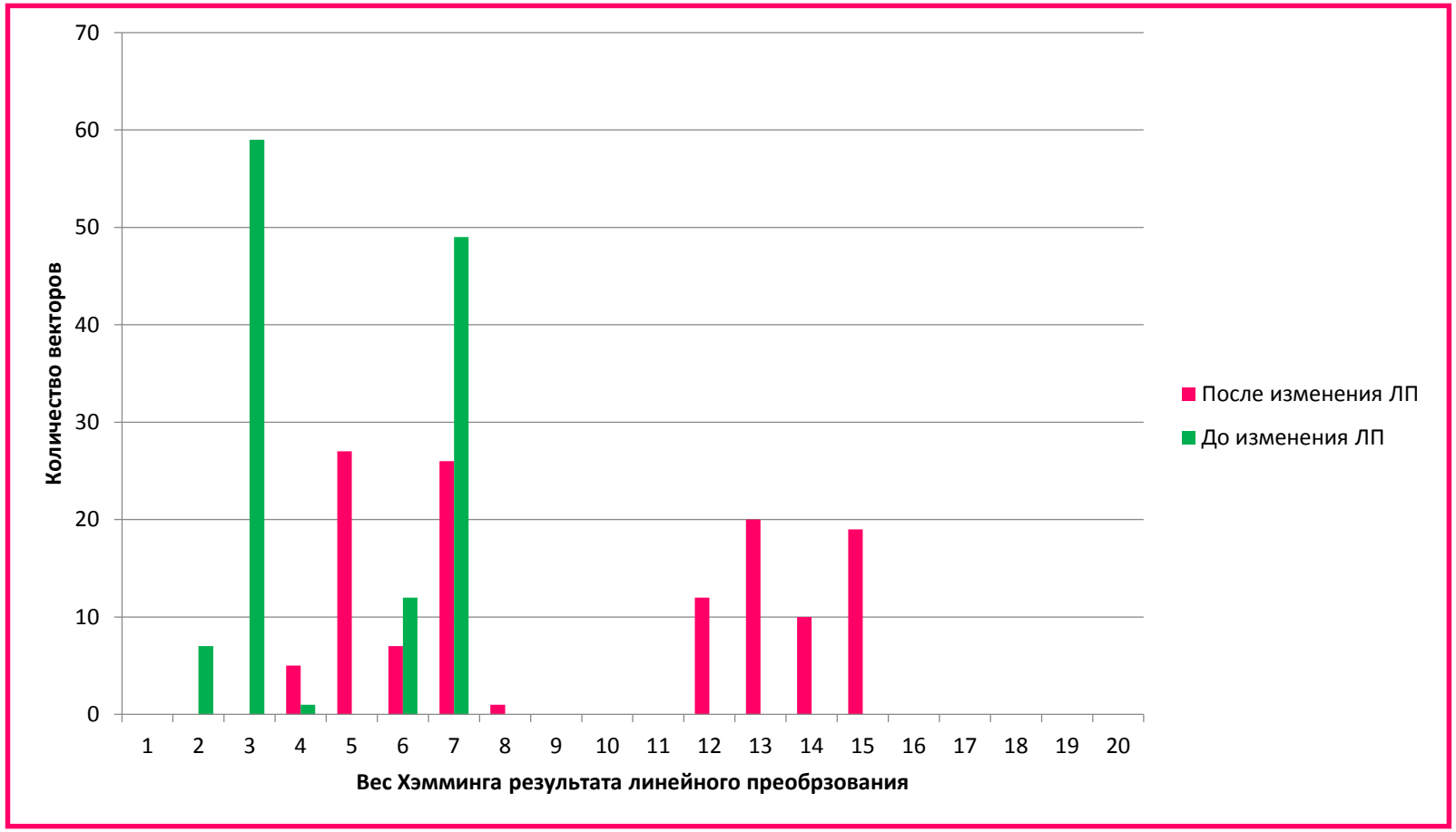
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S0: | 3 | 8 | 15 | 1 | 10 | 6 | 5 | 11 | 14 | 13 | 4 | 2 | 7 | 0 | 9 | 12 |
| S1: | 15 | 12 | 2 | 7 | 9 | 0 | 5 | 10 | 1 | 11 | 14 | 8 | 6 | 13 | 3 | 4 |
| S2: | 8 | 6 | 7 | 9 | 3 | 12 | 10 | 15 | 13 | 1 | 14 | 4 | 0 | 11 | 5 | 2 |
| S3: | 0 | 15 | 11 | 8 | 12 | 9 | 6 | 3 | 13 | 1 | 2 | 4 | 10 | 7 | 5 | 14 |
| S4: | 1 | 15 | 8 | 3 | 12 | 0 | 11 | 6 | 2 | 5 | 4 | 10 | 9 | 14 | 7 | 13 |
| S5: | 15 | 5 | 2 | 11 | 4 | 10 | 9 | 12 | 0 | 3 | 14 | 8 | 13 | 6 | 7 | 1 |
| S6: | 7 | 2 | 12 | 5 | 8 | 4 | 6 | 11 | 14 | 9 | 1 | 15 | 13 | 3 | 10 | 0 |
| S7: | 1 | 13 | 15 | 0 | 14 | 8 | 2 | 11 | 7 | 4 | 12 | 10 | 9 | 3 | 5 | 6 |

# Изменение класса линейных преобразований

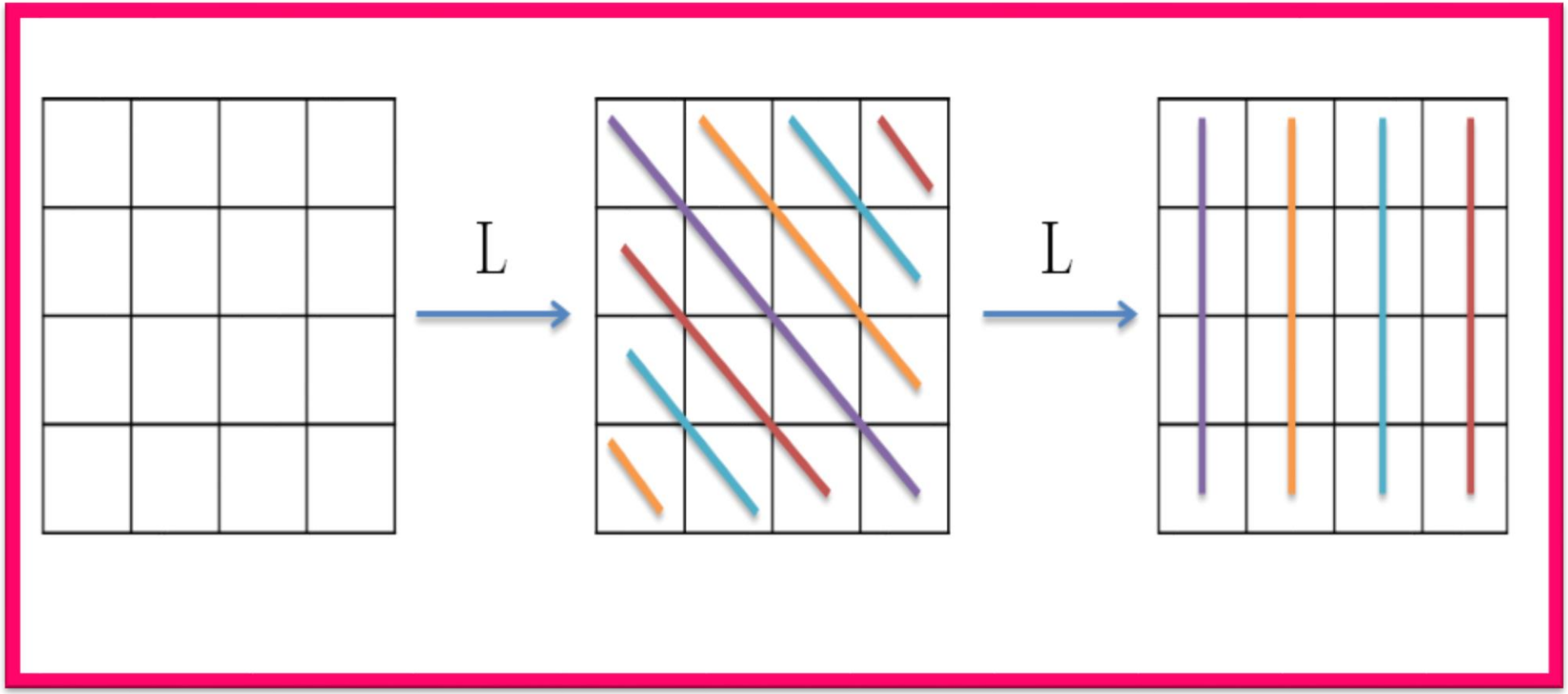
| Класс л/п до изменения | Класс л/п после изменения |
|---|---|
| $x_1 := x_1 \oplus x_2 \oplus x_3$<br>$x_4 := x_4 \oplus x_2 \oplus (x_3 \ll c_1)$<br>$x_1 := (x_1 \lll c_2)$<br>$x_4 := (x_4 \lll c_3)$<br>$x_2 := x_2 \oplus x_1 \oplus x_4$<br>$x_3 := x_3 \oplus x_1 \oplus (x_4 \ll c_4)$ | $x_1 := x_1 \oplus x_2 \oplus x_3$<br>$x_4 := x_4 \oplus x_3 \oplus (x_2 \ll c_1)$<br>$x_1 := (x_1 \lll c_2)$<br>$x_4 := (x_4 \lll c_3)$<br>$x_2 := x_2 \oplus x_1 \oplus x_4$<br>$x_3 := x_3 \oplus x_4 \oplus (x_1 \ll c_4)$<br>$x_2 := (x_2 \lll c_5)$<br>$x_4 := (x_4 \lll c_6)$<br>$x_1 := x_1 \oplus x_2 \oplus (x_4 \ll c_7)$<br>$x_3 := x_3 \oplus x_4 \oplus x_2$ |

# Улучшение рассеивания в линейном преобразовании

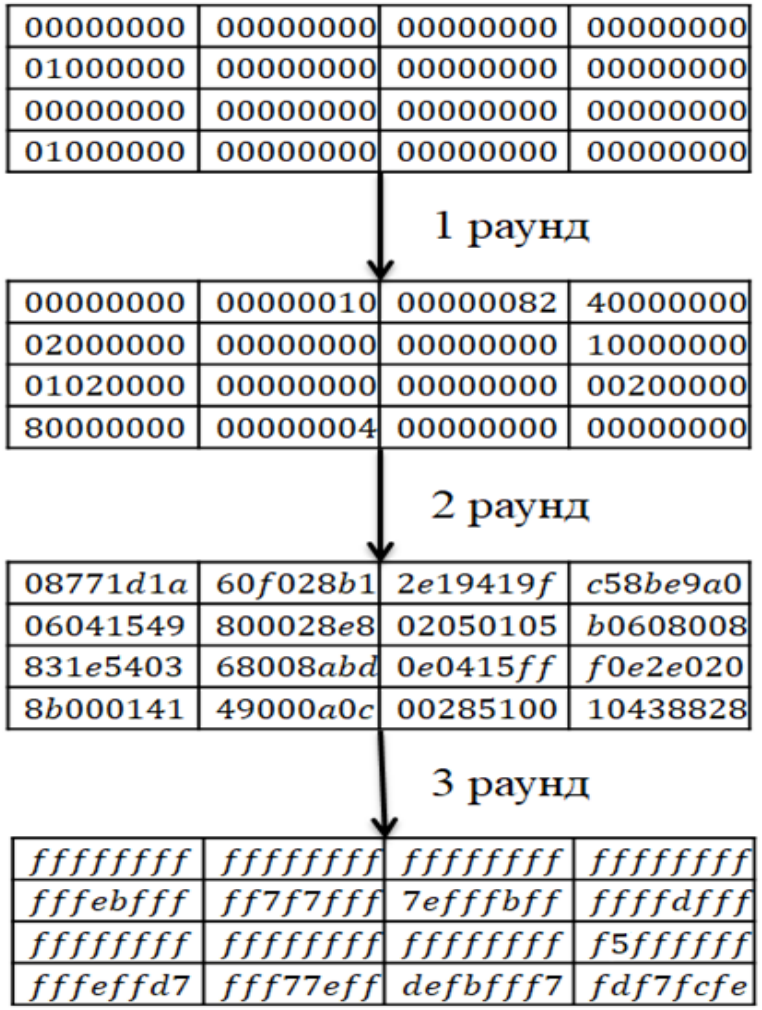
Увеличение влияния одного бита разницы с от 2-7 до от 4-15 бит результата



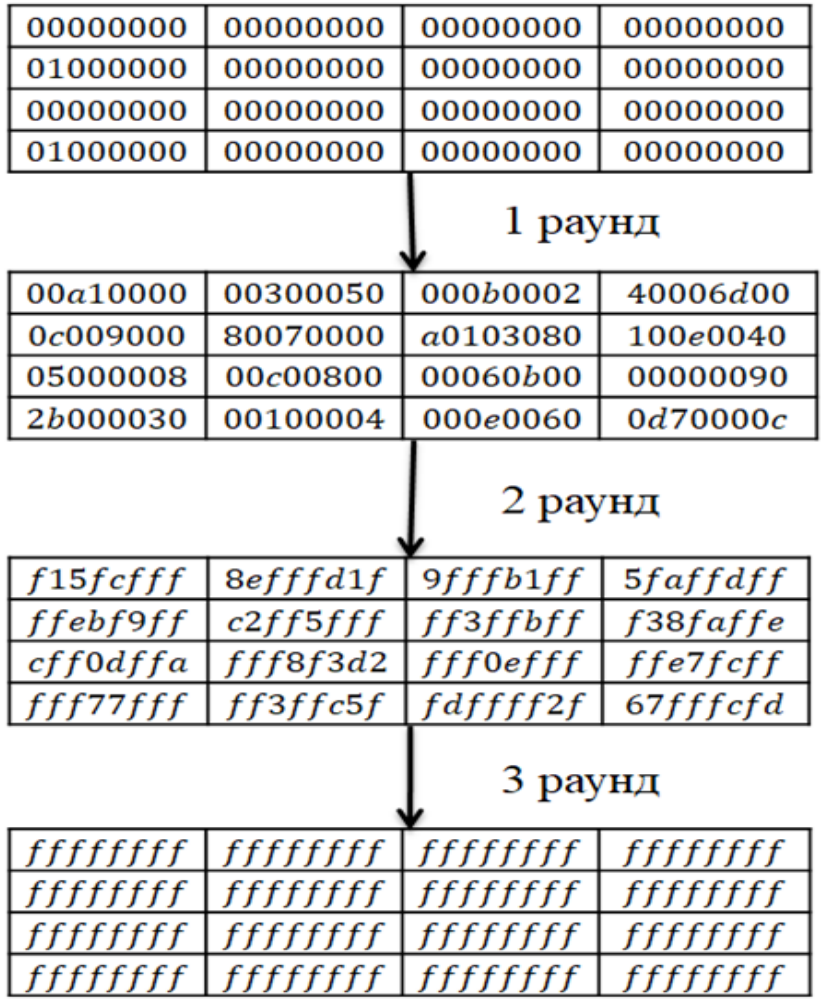
# Изменения слоя линейных преобразований



# Поиск неактивных бит после изменений




До изменений




После изменений

# Особенности реализации алгоритма

Параллельное вычисление:

- Сложения с константными значениями и счетчиком;
- Слой из s-блоков;
- Слоя линейных преобразований.

Реализация линейного преобразования расширения сообщения при помощи использования таблиц поиска.

# Таблицы поиска для расширения сообщения

Общее число таблиц поиска: $\frac{n}{8 \cdot p}$

Объем каждой таблицы в битах: $2^p \cdot n$

Значения для Hamsi-256

| $p$ | объем каждой таблицы | количество таблиц |
|---|---|---|
| **2** | 2048 | 16 |
| **3** | 2688 | 11 |
| **4** | 4096 | 8 |
| **5** | 6272 | 7 |
| **6** | 10368 | 6 |
| **7** | 16896 | 5 |
| **8** | 32768 | 4 |

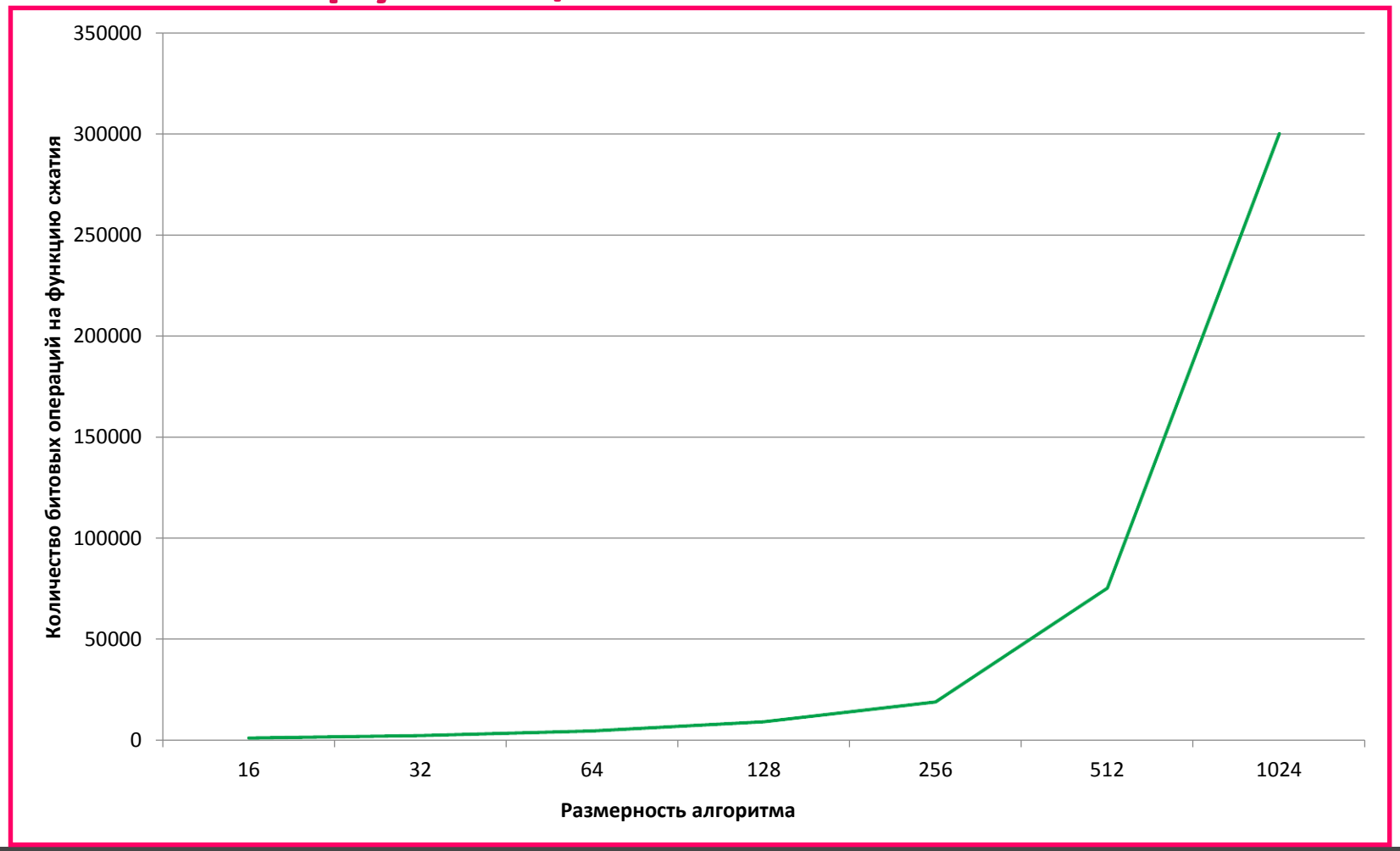
# Оценка количества операций на выполнение функции сжатия

| Преобразование | Количество битовых операций (И,ИЛИ,XOR) |
|---|---|
| Расширения сообщения | $\frac{n^2}{8 \cdot p}$ |
| Конкатенация сообщений | – |
| $x$ раундов нелинейной перестановки P | $23 \cdot n \cdot x$ |
| Усечение сообщения | – |
| Сложение с предыдущим цепным значением | $n$ |

**Общее число битовых операций**

$$n \cdot (23 \cdot x + 1 + \frac{n}{8 \cdot p})$$

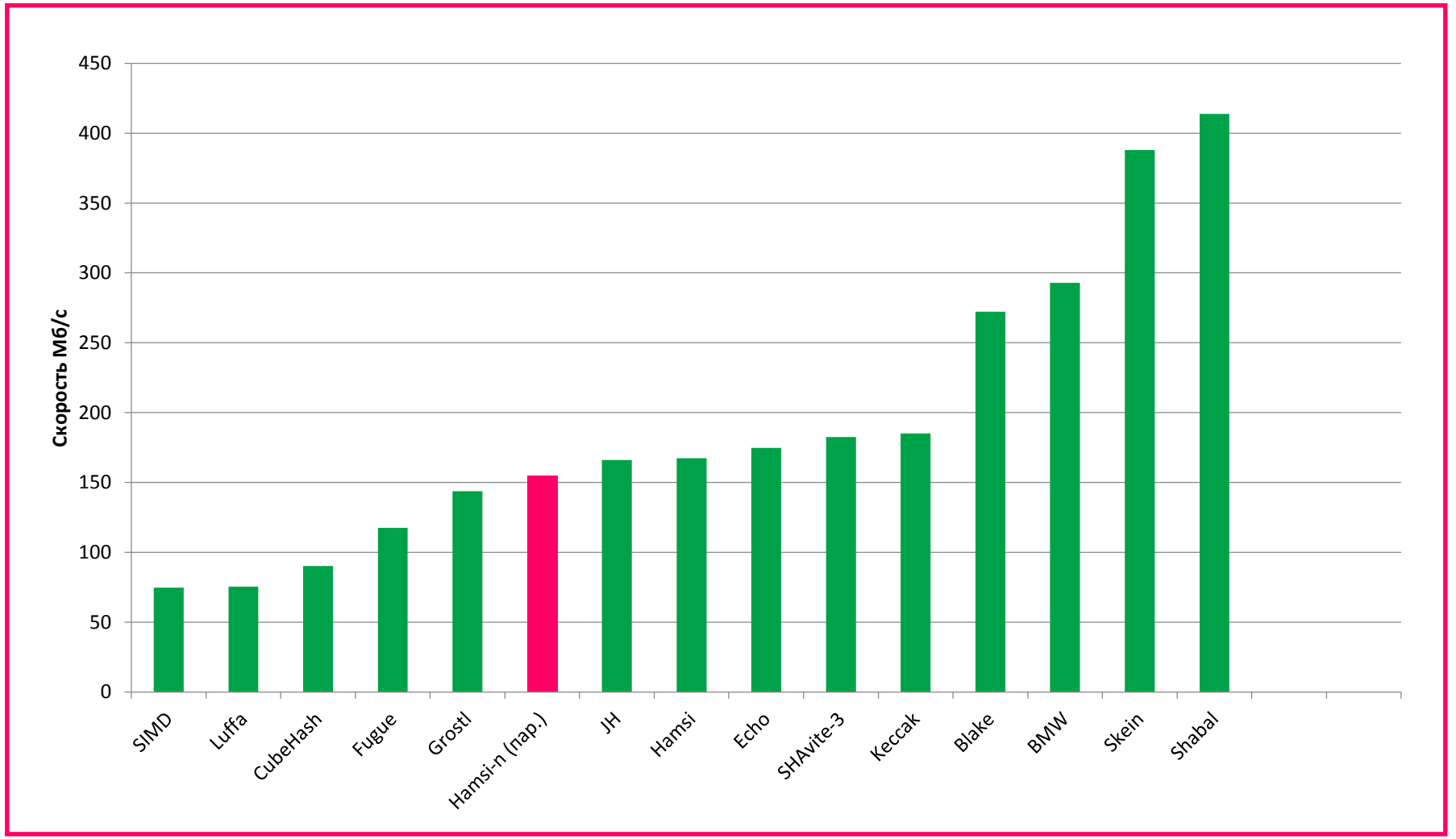
# Оценка количества операций на выполнение функции сжатия



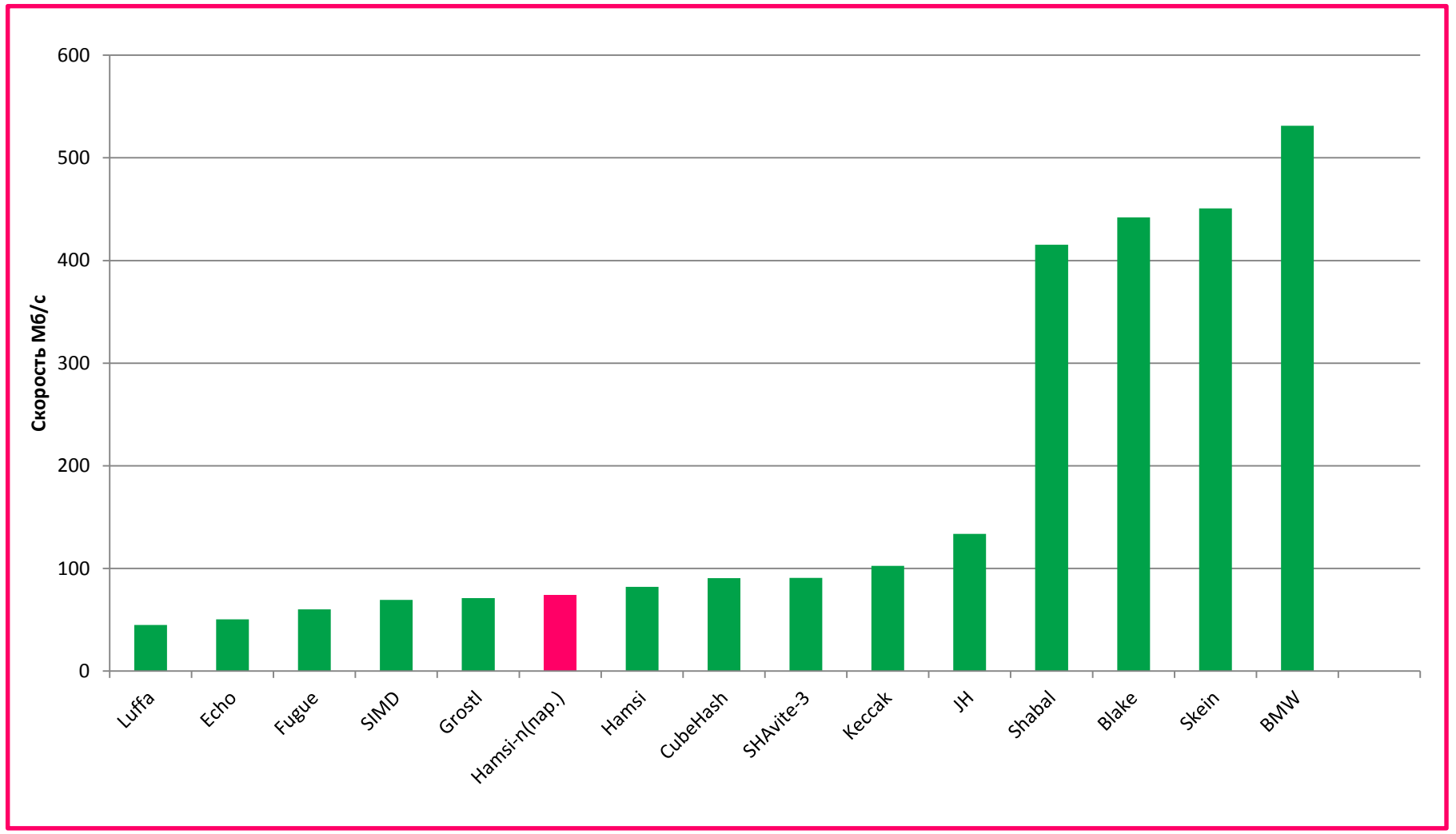
# Сравнение скорости параметризованной версии с другими участниками конкурса SHA-3

| Алгоритм | 256-битная версия | | 512-битная версия | |
|---|---|---|---|---|
| | Мбайт/с | Тактов/байт | Мбайт/с | Тактов/байт |
| Blake | 272,17 | 15,43 | 441,97 | 9,50 |
| BMW | 292,81 | 14,34 | 531,34 | 7,90 |
| CubeHash | 90,28 | 46,52 | 90,49 | 46,41 |
| Echo | 174,71 | 24,34 | 50,31 | 83,48 |
| Fugue | 117,51 | 35,74 | 60,18 | 69,79 |
| Grostl | 143,83 | 29,20 | 71,01 | 59,14 |
| Hamsi | 167,34 | 23,68 | 81,93 | 51,26 |
| **Hamsi-n(пар.)** | **155,03** | **25,44** | **73,72** | **56,97** |
| JH | 166,08 | 30,86 | 133,72 | 31,40 |
| Keccak | 185,17 | 22,68 | 102,37 | 41,02 |
| Luffa | 75,36 | 55,73 | 44,95 | 93,43 |
| SHAvite-3 | 182,59 | 27,52 | 90,64 | 46,33 |
| Shabal | 413,83 | 10,14 | 415,36 | 10,11 |
| SIMD | 74,76 | 56,17 | 69,24 | 60,65 |
| Skein | 388,03 | 10,82 | 450,76 | 9,31 |

# Сравнение скорости 256-битных версий



# Сравнение скорости 512-битных версий



# Заключение

- ✓ В качестве основы алгоритма была взята и проанализирована хеш-функция Hamsi;
- ✓ На основе полученного анализа был описан параметризованный алгоритм Hamsi-$n$;
- ✓ Функция сжатия изменена в соответствии с изученными работами по криптоанализу Hamsi;
- ✓ Осуществлен анализ скорости работы параметризованного алгоритма.

# Спасибо за внимание!

# Контактная информация

Электронная почта:

ermakov-kir@mail.ru

Телефон:

+7 929 039-25-24