УДК 519.719.2 DOI 10.17223/20710410/68/3

СЛОЖНОСТЬ ВЫЧИСЛЕНИЯ НЕКОТОРЫХ ПОДСТАНОВОК, ИМЕЮЩИХ *TU*-ПРЕДСТАВЛЕНИЕ

Д. Б. Фомин*, Д. И. Трифонов**

**Академия криптографии Российской Федерации, г. Москва, Россия*

***Технический комитет по стандартизации «Криптографическая защита информации», г. Москва, Россия*

E-mail: dbfomin@kryptonian.ru, d.arlekino@gmail.com

Рассматривается проблема оценки вычислительной сложности определённых классов подстановок, обладающих *TU*-представлением. В качестве метрик выбраны комбинационная сложность и глубина функции, задающей данную подстановку. Для получения оценок исследуется представление элементов поля в различных базисах: полиномиальном, нормальном, смешанном, а также с использованием PRR- и RRB-представлений элементов поля. Основное внимание уделяется анализу различных представлений элементов поля и их влиянию на вычислительную сложность. Комбинационная сложность оценивается на основе количества элементарных операций, необходимых для реализации подстановки; глубина функции определяется как максимальное количество логических уровней в схеме. Изучение различных базисов позволяет выявить наиболее эффективные способы представления, способствующие минимизации вычислительной сложности. В качестве примера приводится оценка указанных характеристик для подстановки пространства $\mathbb{F}_2^8$, используемой в отечественных стандартизированных симметричных криптографических алгоритмах. Получена минимальная из известных оценка комбинационной сложности, равная 169.

Ключевые слова: *подстановка, комбинационная сложность, глубина функции, конструкция типа «бабочка», TU-представление.*

COMPUTATIONAL WORK FOR SOME *TU*-BASED PERMUTATIONS

D. B. Fomin*, D. I. Trifonov**

**The Academy of Cryptography of the Russian Federation, Moscow, Russia*

***Technical Committee "Cryptography and Security Mechanism", Moscow, Russia*

The problem of evaluating the computational complexity of certain classes of substitutions with a *TU*-representation is considered. The metrics used include combinatorial complexity and the depth of the function that defines the substitution. To obtain these evaluations, the representation of field elements in various bases is investigated, including polynomial, normal, mixed, as well as PRR and RRB representations. The primary focus is on analyzing different representations of field elements and their impact on computational complexity. The combinatorial complexity is assessed based on the number of elementary operations required to implement the substitution, while the function depth is determined by the maximum number of logical levels in the circuit. The use of different bases allows us to identify the most effective representation methods that help minimize computational complexity. As an example, we

provide an evaluation of the specified characteristics for the substitution used in Russian standardized symmetric algorithms. The lowest known estimate of combinatorial complexity has been obtained, which equals 169.

Keywords: *permutation, combinational complexity, circuit depth, butterfly, TU-decomposition.*

Введение

Сложность вычислений является одной из ключевых проблем, актуальных в современных исследованиях в области вычислительной науки. Увеличение объёма обрабатываемых данных, широкое внедрение технологий Интернета вещей (IoT) и машинной связи (M2M) подчеркивает значимость разработки высокоэффективных и безопасных систем. Оценка сложности вычислений имеет большое значение как с теоретической, так и с практической точки зрения, позволяя, с одной стороны, понять фундаментальные ограничения и возможности алгоритмов, с другой — производить оптимизацию времени вычислений и необходимых ресурсов для реальных систем.

В настоящее время разработаны подходы к созданию стойких симметричных криптографических алгоритмов, однако вопросы, касающиеся сложности их реализации, часто остаются недостаточно изученными. Одним из ключевых примитивов современных криптографических алгоритмов являются нелинейные биективные преобразования — подстановки. В условиях ограничений современных вычислительных устройств, помимо естественных требований, связанных с их криптографическими характеристиками, возникают дополнительные ограничения, относящиеся к сложности их реализации. В данной работе представлены понятия сложности вычислений, приведён обзор методов построения нелинейных биективных преобразований с низкой вычислительной сложностью, а также рассмотрены подходы к минимизации сложности вычислений подстановок специального вида.

1. Способы построения низкоресурсных нелинейных биективных преобразований с заданными криптографическими свойствами

Существуют три основных подхода к построению подстановок с заданными эксплуатационными характеристиками: полный поиск с использованием обхода графа в глубину и метода встречи посередине [1–3], эвристические методы [3–7] и использование «простых алгебраических конструкций», например мономиальных подстановок (в частности, обращения ненулевых элементов поля) [8]. Однако практически все из этих подходов позволяют оценивать только количество операций. Для того чтобы оценить реальную физическую трудоёмкость реализации, предлагается реализовывать узлы на реальных физических устройствах [9] или использовать знание о трудоёмкости реализации каждой базисной функции для эвристического поиска оптимальной реализации [3].

Фундаментальной монографией в области оценки вычислительной сложности можно считать работу Д. Э. Сэвиджа [10]. Введённые им метрики сложности позволяют оценивать эффективность реализации на различных платформах при сохранении достаточного уровня математической строгости.

При описании представления реализации произвольной функции $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ зачастую используются логические схемы [11]. Логическая схема, которую мы, согласно [10], также будем называть комбинационной машиной, представляет собой соединение элементов некоторого базиса Ω, каждый из которых реализует некоторую логиче-

скую (булеву) функцию указанного базиса. Логическая схема может быть представлена в виде ориентированного графа, при этом вершины, имеющие полустепень захода равную нулю, обозначают аргументы функции f, а вершины, имеющие полустепень исхода равную нулю, — значение функции f.

Определение 1. Комбинационная сложность функции $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ в базисе Ω, обозначаемая $C_\Omega(f)$, есть минимальное число элементов базиса Ω, достаточное для реализации функции f логической схемой.

Определение 2. Глубина функции $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ в базисе Ω, обозначаемая $D_\Omega(f)$, есть число логических элементов, расположенных на самом длинном ориентированном пути графа, представляющего логическую схему.

В данной работе под мерой сложности функций f понимаются комбинационная сложность и глубина функции в некотором базисе.

Замечание 1. В качестве базиса Ω будем использовать $\Omega = \{\wedge, \vee, \oplus, \neg\}$; в случае, когда это понятно из контекста, будем опускать его и говорить просто о «комбинационной сложности функции f» и «глубине функции f».

Представляет интерес задача нахождения для заданной функции f логической схемы «минимального размера», то есть задача вычисления её комбинационной сложности. Хорошо известны разработанные для этих целей метод карт Карно и его обобщение — процедура Квайна — Мак-Класки [12]. Применение этого метода позволяет минимизировать размер схем в случае реализации функций формулами, имеющими вид суммы произведений (дизъюнктивной нормальной формы). О. Б. Лупановым показано [13], что функция сложения по модулю 2 (функция чётности, равная единице, если нечётно число единиц среди её аргументов $x_1, \ldots, x_n$, где $x_i \in \{0, 1\}$, $i = 1, \ldots, n$) реализуется при указанных ограничениях схемой экспоненциального (относительно n) размера, тогда как при отсутствии ограничений возможна реализация схемами линейного размера. Аналогичные результаты справедливы для некоторых других функций.

Определение 3 [14]. Пусть $F\colon \mathbb{F}_2^{n-t} \times \mathbb{F}_2^t \to \mathbb{F}_2^{n-t} \times \mathbb{F}_2^{m-n+t}$; $m - n + t \geqslant 1$; $x_1, y_1 \in \mathbb{F}_2^{n-t}$; $x_2 \in \mathbb{F}_2^t$; $y_2 \in \mathbb{F}_2^{m-n+t}$; $T\colon \mathbb{F}_2^{n-t} \times \mathbb{F}_2^t \to \mathbb{F}_2^{n-t}$; $U\colon \mathbb{F}_2^t \times \mathbb{F}_2^{n-t} \to \mathbb{F}_2^{m-n+t}$. Если функция F имеет представление

$$F(x_1, x_2) = (y_1, y_2) = \big(T(x_1, x_2), U(x_2, T(x_1, x_2))\big), \tag{1}$$

где $T(x_1, x_2)$ является биективным отображением по x_1 при фиксированном значении $x_2 \in \mathbb{F}_2^{n-t}$, то такое представление функции F в виде (1) называется TU-представлением (рис. 1).

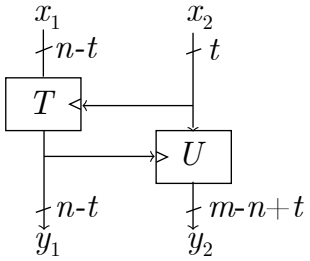

Рис. 1. Графическое изображение функции, имеющей TU-представление

2. Способы представления элементов поля

Пусть $\mathbb{F}_{2^n}$ — конечное поле из 2^n элементов. В нём есть подполе $\mathbb{F}_2$, что позволяет рассматривать поле $\mathbb{F}_{2^n}$ как векторное пространство над полем $\mathbb{F}_2$, имеющее некоторый базис $\alpha_1, \alpha_2, \ldots, \alpha_n$, состоящий из n элементов. Таким образом, для любого поля $\mathbb{F}_{2^n}$, зафиксировав базис, каждый его элемент естественным образом можно представить в виде вектора элементов поля $\mathbb{F}_2$ длины n, что позволяет задать однозначное отображение $\sigma\colon \mathbb{F}_{2^n} \to V_n$ из множества элементов поля в множество вектор-строк.

Для произвольного k, такого, что $k|n$, в поле $\mathbb{F}_{2^n}$ существует подполе из 2^k элементов, которое обозначим $\mathbb{F}_{2^k}$. При этом существует неприводимый многочлен $f(x)$ степени $m = n/k$ над $\mathbb{F}_{2^k}$, такой, что $\mathbb{F}_{2^n} \cong \mathbb{F}_{2^k}[x]/f(x)$ и $[x]_{f(x)}$ есть корень многочлена $f(x)$ в поле $\mathbb{F}_{2^k}[x]/f(x)$.

Определение 4. Пусть α — элемент поля $\mathbb{F}_{2^n}$, такой, что множество $\{\alpha^i : i = 0, \ldots, m-1\}$ является базисом $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$. Говорят, что $\{\alpha^i : i = 0, \ldots, m-1\}$ является полиномиальным базисом поля $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$; элемент α называется образующим полиномиального базиса.

Полиномиальный базис будем обозначать Poly. Помимо термина «полиномиальный базис», в литературе можно встретить эквивалентные названия: «стандартный базис» и «канонический базис». Очевидно, что $\left\{[x]^i_{f(x)} : i = 0, \ldots, m-1\right\}$ является полиномиальным базисом поля $\mathbb{F}_{2^k}[x]/f(x)$ над $\mathbb{F}_{2^k}$. Элемент α является образующим полиномиального базиса тогда и только тогда, когда α — корень многочлена $f(x)$ над $\mathbb{F}_{2^n}$. Если α — корень неприводимого над $\mathbb{F}_{2^k}$ многочлена степени m, то все корни в поле $\mathbb{F}_{2^n}$ есть элементы множества $\left\{\alpha^{2^{ki}} : i = 0, \ldots, m-1\right\}$, и они линейно независимы над $\mathbb{F}_{2^k}$.

Определение 5. Пусть α — элемент поля $\mathbb{F}_{2^n}$, такой, что множество $\left\{\alpha^{2^{ki}} : i = 0, \ldots, m-1\right\}$ является базисом $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$. Говорят, что $\left\{\alpha^{2^{ki}} : i = 0, \ldots, m-1\right\}$ является нормальным базисом поля $\mathbb{F}_{2^n}$ над $\mathbb{F}_{2^k}$; элемент α называется образующим нормального базиса.

Нормальный базис будем обозначать Norm. Очевидно, что для поля $\mathbb{F}_{2^n}$ существует как минимум один нормальный базис и корень α неприводимого над $\mathbb{F}_{2^k}$ многочлена $f(x)$ в поле $\mathbb{F}_{2^n}$ является образующим как полиномиального, так и нормального базиса.

Говоря о подстановке на множестве элементов поля, будем подразумевать, что подстановка действует на векторное представление элементов поля.

Известно, что полиномиальный и нормальный базисы эффективны для реализации умножения и эндоморфизма Фробениуса соответственно [15]. При этом в настоящее время наиболее эффективный способ уменьшения комбинационной сложности и глубины функций, реализующих операции в поле, заключается в использовании теоремы о башне полей и реализации операций в подполе. Например, в работе [8] предлагается элементы поля $\mathbb{F}_{2^8}$ представлять в виде вектора $\mathbb{F}^2_{2^4}$, а элементы поля $\mathbb{F}_{2^4}$ рассматривать в виде вектора $\mathbb{F}^2_{2^2}$ и т. д. Таким образом, элементы поля $\mathbb{F}_{2^8}$ представляются векторами из множества $\left(\left(\mathbb{F}_2^2\right)^2\right)^2$.

Помимо описанных представлений элементов поля, существуют и другие способы представления, позволяющие достигать «низких» значений комбинационной сложности и глубины схем для функций, реализующих операции в поле. В работе [16] предлагается задавать элементы поля $\mathbb{F}_{2^n}$ с использованием $m \geqslant n$ бит следующим образом. Пусть $f(x)$ — неприводимый многочлен степени n над полем $\mathbb{F}_2$; многочлен $g(x)$ сте-

пени $m-n$ над полем $\mathbb{F}_2$ такой, что НОД$(f(x), g(x)) = 1$ и $g(0) \neq 0$. Тогда можно рассмотреть многочлен $p(x) = g(x)f(x)$ степени m.

Будем задавать элементы поля $\mathbb{F}_2^n$ многочленами степени не больше m таким образом, чтобы они образовывали подпространство размерности n в векторном пространстве размерности m (фактически задаётся CRC-(m,n)-код). Элементы поля $\mathbb{F}_{2^n}$ однозначно определяются элементами фактор-кольца многочленов по модулю $p(x)$, которые делятся на $g(x)$ без остатка. Заметим, что операция умножения таких многочленов корректно определена и задаётся через умножение по модулю многочлена $p(x)$. В зарубежной литературе такое представление называется Polynomial Ring Representation или PRR [16].

Согласно [17], сложность операций в поле, как правило, тем меньше, чем меньше коэффициентов в записи неприводимого многочлена, его задающего. Таким образом, в случае $p(x) = x^{n+1}+1$ сложность операций потенциально снижается. Очевидно, многочлен $x^{n+1}+1$ не является неприводимым, что не позволяет задавать с его помощью поле. Однако он может быть использован для реализации PRR-представления. В [18] предлагается рассмотреть следующий случай:

$$x^{n+1}+1 = (x+1)\left(x^n + x^{n-1} + \ldots + 1\right),$$

где $g(x) = x+1$; $f(x) = x^n + x^{n-1} + \ldots + 1$. Использование неприводимого многочлена $f(x)$ такого вида позволяет эффективно реализовывать операцию умножения в поле, а также операцию возведения в квадрат произвольное число раз, которая есть просто перестановка коэффициентов базисных векторов [16].

Заметим, что в PRR-представлении каждый элемент поля однозначно представляется одним элементом фактор-кольца $\mathbb{F}_2[x]/p(x)$. Однако с использованием всех элементов фактор-кольца $\mathbb{F}_2[x]/p(x)$ можно задать элементы кольца. В этом случае одному элементу поля можно поставить в соответствие несколько элементов фактор-кольца. Действительно, если $t_1, t_2 \in \mathbb{F}_2[x]/p(x)$ и $t_1(x) = t_2(x) \pmod{f(x)}$, то t_1 и t_2 задают один элемент поля $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/f(x)$. В случае, когда $p(x) = x^{n+1}+1 = (x+1)(x^n + x^{n-1} + \ldots + 1)$, такое представление в зарубежной литературе носит название RRB-представления [19]. Известно, что его использование уменьшает глубину операции умножения элементов поля [18].

Замечание 2. В данной работе оцениваются комбинационные сложности и глубина функций, реализующих некоторые функции, задаваемые над полем. Так как вид функции напрямую зависит от базиса и поля, над которым рассматривается преобразование, то введём дополнительные обозначения: через $C_\Omega(f; \mathbb{F}, \text{Basis})$ и $D_\Omega(f; \mathbb{F}, \text{Basis})$ будем обозначать соответственно комбинационную сложность и глубину функции f, определённой над полем $\mathbb{F}$ в базисе Basis. При использовании башни полей в качестве базиса будем указывать базис расширения.

3. Некоторые классы нелинейных биективных преобразований

В работе [20] впервые были предложены классы подстановок пространства $\mathbb{F}_2^8$, имеющих TU-представление и обладающие «высокими» показателями криптографических характеристик. В работах [20–22] данные подстановки были обобщены и предложены параметрические семейства нелинейных биективных преобразований, а также оценены их криптографические характеристики.

Определение 6. Пусть $x_1, x_2 \in \mathbb{F}_2^m$, $\pi_i, \widehat{\pi}_i \in S(\mathbb{F}_2^m)$, $\pi_i(0) = 0$, $\widehat{\pi}_i(0) = 0$, $i = 1, 2$. Тогда подстановку $F_{\text{А}}(x_1, x_2) = (y_1, y_2)$, определяемую равенствами

$$y_1 = \begin{cases} \pi_1(x_1)\, x_2, & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases}$$

$$y_2 = \begin{cases} \pi_2\left((x_2)^2\, \pi_1(x_1)\right), & y_1 \neq 0, \\ \widehat{\pi}_2(x_2), & y_1 = 0, \end{cases}$$

будем называть подстановкой из параметрического семейства типа «А» или просто подстановкой типа «А» (рис. 2).

Определение 7. Пусть $x_1, x_2 \in \mathbb{F}_2^m$, $\pi_i, \widehat{\pi}_i \in S(\mathbb{F}_2^m)$, $\pi_i(0) = 0$, $\widehat{\pi}_i(0) = 0$, $i = 1, 2$. Тогда подстановку $F_{\text{Б}}(x_1, x_2) = (y_1, y_2)$, определяемую равенствами

$$y_1 = \begin{cases} x_1 \cdot \pi_1(x_2), & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases}$$

$$y_2 = \begin{cases} x_2 \cdot \pi_2\left(x_1 \cdot \pi_1(x_2)\right), & y_1 \neq 0, \\ \widehat{\pi}_2(x_2), & y_1 = 0, \end{cases}$$

будем называть подстановкой из параметрического семейства типа «Б» или просто подстановкой типа «Б» (рис. 3).

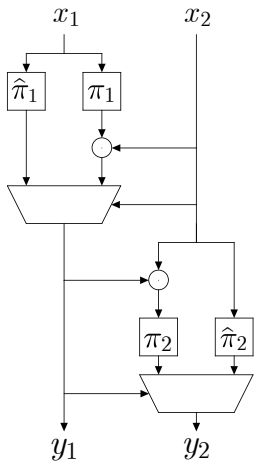

Рис. 2. Подстановка типа «А»

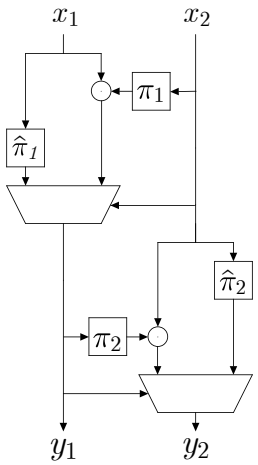

Рис. 3. Подстановка типа «Б»

Рассмотрим семейство подстановок, параметрами которого являются четвёрка степеней $(\alpha, \beta, \gamma, \delta)$ и подстановки $\widehat{\pi}_i \in S(\mathbb{F}_2^m)$, такие, что $\widehat{\pi}_i(0) = 0$, $i = 1, 2$:

$$\begin{aligned} G_1(x_1, x_2) = y_1 &= \begin{cases} x_1^{\alpha} \cdot x_2^{\beta}, & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases} \\ G_2(x_1, x_2) = y_2 &= \begin{cases} x_1^{\gamma} \cdot x_2^{\delta}, & y_1 \neq 0, \\ \widehat{\pi}_2(x_2), & y_1 = 0. \end{cases} \end{aligned} \tag{2}$$

Чтобы (2) задавало биективное преобразование, достаточно, чтобы система уравнений

$$\begin{cases} G_1(x_1, x_2) = a_1, \\ G_2(x_1, x_2) = a_2 \end{cases}$$

имела решение для произвольных $a_1, a_2 \in \mathbb{F}_2^m$. Такое семейство подстановок будем называть подстановкой из параметрического семейства типа «Г», произвольную подстановку из которого будем обозначать F_Γ. Очевидно, что если в качестве параметров в параметрических семействах типов «А» и «Б» выбираются мономиальные подстановки, они представляются подстановками из параметрического семейства типа «Г». В [23] показано, что эта подстановка также имеет TU-представление. Однако такое задание подстановки потенциально позволяет уменьшить глубину схемы из функциональных элементов, реализующей её. Более того, среди всех известных подстановок из предложенных семейств подстановка $G_\Gamma(x_1, x_2) = (y_1, y_2)$ из параметрического семейства «Г» имеет минимальное количество нелинейных преобразований (две подстановки, два умножения и два мультиплексора) и задаётся следующим образом (здесь и далее под подстановкой обращения ненулевых элементов поля $\mathbb{F}_2^n$, $n \geqslant 2$, понимаем подстановку, задаваемую формулой x^{2^n-2}, и обозначаем её x^{-1}):

1) $x' = x_1^{-1}$;
2) $y' = x_2^{-1}$;
3) $x'' = x_1 \cdot y'$;
4) $y'' = x' \cdot y'$;
5) если $x_1 = 0$, то $y_2 = y'$, иначе $y_2 = y''$;
6) если $x_2 = 0$, то $y_1 = x'$, иначе $y_1 = x''$.

Результаты работы [24] показывают эффективность реализации определённых выше нелинейных биективных преобразований на аппаратных платформах с использованием программируемых логических интегральных схем (ПЛИС). Становится актуальной задача оценки комбинационной сложности и глубины функции для подстановок из представленных семейств, что важно при их программной реализации (bitslice implementation [1, 25, 26]) и аппаратной реализации на сверхбольших интегральных схемах (СБИС) и СБИС с программируемой архитектурой.

Для задания указанных подстановок используются следующие функции: операция умножения в поле; мультиплексор (условный выбор); подстановки.

Для параметрических семейств типов «А» и «Б» в [20, 22] в качестве подстановок π_1, π_2 рассматриваются мономиальные подстановки. В данной работе также будем рассматривать в качестве указанных параметров мономиальные подстановки.

Подстановки $\widehat{\pi}_1, \widehat{\pi}_2$ параметрических семейств «А», «Б» и «Г» в работе [27] предлагается выбирать с использованием эвристического алгоритма.

Замечание 3. Проведено экспериментальное исследование классов аффинной эквивалентности подстановок $\widehat{\pi}_1, \widehat{\pi}_2$ для рассматриваемых параметрических семейств в случае построения подстановок пространства $\mathbb{F}_2^8$ с помощью алгоритма из [27]. Это можно сделать, так как для подстановок пространства $\mathbb{F}_2^4$ имеется полная их аффинная классификация [28]. Результаты экспериментов показали, что в подавляющем большинстве случаев (более 97 %) указанные подстановки принадлежат двум семействам подстановок $\mathbb{F}_2^4$ с представителями x^{14} и $x^7 + x^4 + x$. Таким образом, представляет интерес нахождение сложности реализации этих подстановок.

Определение 8. Две подстановки $F, G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ называются аффинно эквивалентными, если существует пара невырожденных аффинных преобразований $A_1, A_2 \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, таких, что $G(x) = A_2(F(A_1(x)))$ для всех $x \in \mathbb{F}_2^n$. Аффинное преобразование — это отображение вида $A(x) = Mx + b$, где $M \in \mathrm{GL}(n, 2)$ — невырожденная матрица; $b \in \mathbb{F}_2^n$ — вектор.

Замечание 4. Помимо рассматриваемых в данной работе подстановок, подход, изложенный далее, применим для ряда подстановок, также имеющих TU-представление, например, для подстановки пространства $\mathbb{F}_2^8$, используемой в отечественных стандартизированных симметричных алгоритмах [29] (рис. 4), а также подстановки из работы [30] (рис. 5).

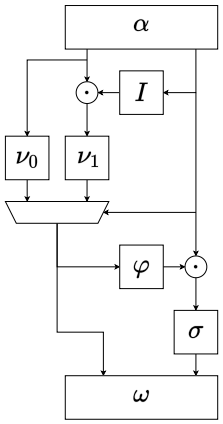

Рис. 4. Подстановка алгоритма «Кузнечик» [29]

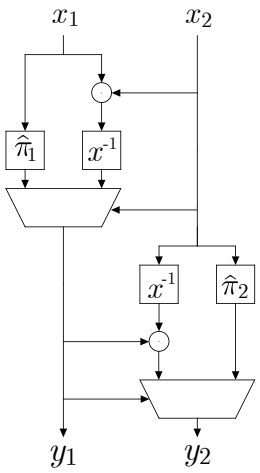

Рис. 5. Подстановка из работы [30]

4. Сложность реализации некоторых классов нелинейных биективных преобразований

4.1. Сложность задания функций над полем $\mathbb{F}_{2^2}$ в нормальном базисе

Поле $\mathbb{F}_{2^2}$ задается единственным неприводимым многочленом второй степени $x^2 + x + 1$ над полем $\mathbb{F}_2$. Для того чтобы построить поле $\mathbb{F}_{(2^2)^2}$, необходимо выбрать неприводимый многочлен степени 2 над полем $\mathbb{F}_{2^2}$.

Известно, что многочлен $x^2+x+\varepsilon$ неприводим над полем $\mathbb{F}_{2^n}$ тогда и только тогда, когда $\mathrm{tr}(\varepsilon) = 1$, где через $\mathrm{tr}_{\mathbb{F}_q}^{\mathbb{F}_{q^m}} : \mathbb{F}_{q^m} \to \mathbb{F}_q$ (или просто tr) обозначен след из поля $\mathbb{F}_{q^m}$ в поле $\mathbb{F}_q$, ставящий в соответствие произвольному элементу $\alpha \in \mathbb{F}_{q^m}$ элемент $\mathrm{tr}_{\mathbb{F}_q}^{\mathbb{F}_{q^m}}(\alpha) = \alpha + \alpha^q + \ldots + \alpha^{q^{m-1}}$ [31]. Произвольный многочлен $f(x) = ax^2 + bx + c$, у которого $a, b \neq 0$, можно привести к этой форме, выполнив преобразование $(a/b^2)f(bx/a)$. Будем в дальнейшем рассматривать только многочлены такого вида.

Здесь и далее, если не сказано иное, элементы поля $\mathbb{F}_2$ будем обозначать курсивным шрифтом a, b, c, элементы поля $\mathbb{F}_{2^2}$ — прямым a, b, c, базисные векторы — греческими буквами α, β, γ, элементы поля $\mathbb{F}_{(2^2)^2}$ — жирным прямым шрифтом $\mathbf{a}, \mathbf{b}, \mathbf{c}$, а его базисные векторы — греческими жирными буквами $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$.

В работе [32] предлагается следующий способ реализации операций в поле $\mathbb{F}_{2^2}$. Пусть $e(x) = x^2 + x + 1$ и его корнем является элемент α. Тогда $\alpha^2 + \alpha = 1$ и $\alpha^3 = 1$, а также $\alpha^3 = \alpha^2 + \alpha$. Рассмотрим нормальный базис $\{\alpha, \alpha^2\}$. Следующие результаты напрямую следуют из работы [32].

Предложение 1. Пусть «$\cdot$» — операция умножения в $\mathbb{F}_{2^2}$, тогда для $x, y \in \mathbb{F}_{2^2}$

$$C_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 7, \quad D_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 3.$$

Действительно, согласно [32]: пусть $\mathrm{a} = a_0\alpha + a_1\alpha^2$, $\mathrm{b} = b_0\alpha + b_1\alpha^2$, $\mathrm{c} = c_0\alpha + c_1\alpha^2$, $\mathrm{a}\cdot\mathrm{b} = \mathrm{c}$. Тогда

$$\mathrm{a}\cdot\mathrm{b} = \left(a_0\alpha + a_1\alpha^2\right)\left(b_0\alpha + b_1\alpha^2\right) =$$
$$= \left((a_0+a_1)(b_0+b_1) + a_0b_0\right)\alpha + \left((a_0+a_1)(b_0+b_1) + a_1b_1\right)\alpha^2 = c_0\alpha + c_1\alpha^2.$$

Предложение 2. Пусть α, α^2 — элементы поля $\mathbb{F}_{2^2}$, задающие его нормальный базис, тогда для $x \in \mathbb{F}_{2^2}$ имеет место: $C_\Omega(x\cdot\alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = C_\Omega(x\cdot\alpha^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$, $D_\Omega(x\cdot\alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = D_\Omega(x\cdot\alpha^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$.

Доказательство следует из формул [32]

$$\alpha\mathrm{a} = a_1\alpha + (a_0+a_1)\alpha^2,\ \ \alpha^2\mathrm{a} = (a_0+a_1)\alpha + a_0\alpha^2.$$

Предложение 3. Для $x \in \mathbb{F}_{2^2}$ выполняется

$$C_\Omega(x^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 0, \quad D_\Omega(x^2; \mathbb{F}_{2^2}; \mathrm{Norm}) = 0.$$

Верность предложения 3 следует из двух ключевых наблюдений. Во-первых, операция возведения в квадрат является эндоморфизмом Фробениуса. Во-вторых, в $\mathbb{F}_{2^2}$ выполнено равенство $x^2 = x^{-1}$.

4.2. Сложность задания функций над полем $\mathbb{F}_{(2^2)^2}$ в полиномиальном базисе

В работе [33] поле $\mathbb{F}_{(2^2)^2}$ предлагается рассматривать в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$, который будем обозначать Poly. Поле $\mathbb{F}_{(2^2)^2}$ строится с использованием неприводимого многочлена $g(x) = x^2 + x + \alpha$, где α является базисным элементом поля $\mathbb{F}_{2^2}$ в нормальном базисе. Аналогично сказанному выше, неприводимость многочлена $g(x)$ следует из [31, следствие 3.79, с. 163] и того факта, что $\mathrm{tr}(\alpha) \neq 0$.

Приведём некоторые результаты, следующие из [33]. Пусть $\mathbf{a} = \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}$, $\mathbf{b} = \mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta}$, $\mathrm{a}_i, \mathrm{b}_i \in \mathbb{F}_{2^2}$, $i = 1, 2$.

Предложение 4. Пусть «$\cdot$» — операция умножения в поле $\mathbb{F}_{(2^2)^2}$, задаваемом неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x, y \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x\cdot y; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 30$, $D_\Omega\left(x\cdot y; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 5$.

Доказательство. Полное доказательство приведено для наглядности. Для последующих предложений справедлива такая же схема доказательства.

Рассмотрим два элемента $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{(2^2)^2}$, представимые в виде

$$\mathbf{a} = \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}, \quad \mathbf{b} = \mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta},$$

где $\mathrm{a}_0, \mathrm{a}_1, \mathrm{b}_0, \mathrm{b}_1 \in \mathbb{F}_{2^2}$ и представлены в нормальном базисе; $\boldsymbol{\beta}$ — корень неприводимого многочлена $g(x)$, удовлетворяющий соотношению $\boldsymbol{\beta}^2 = \boldsymbol{\beta} + \alpha$ (так как $g(\boldsymbol{\beta}) = 0$).

Произведение $\mathbf{c} = \mathbf{a}\cdot\mathbf{b}$ раскрывается следующим образом:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})(\mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta}) = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_0\mathrm{b}_1\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_0\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_1\boldsymbol{\beta}^2.$$

Подставляем $\boldsymbol{\beta}^2 = \boldsymbol{\beta} + \alpha$:

$$\mathbf{a}\cdot\mathbf{b} = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_0\mathrm{b}_1\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_0\boldsymbol{\beta} + \mathrm{a}_1\mathrm{b}_1(\boldsymbol{\beta} + \alpha).$$

Группируем члены с $\boldsymbol{\beta}$ и свободные члены:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha) + (\mathrm{a}_0\mathrm{b}_1 + \mathrm{a}_1\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1)\boldsymbol{\beta}.$$

Обозначим коэффициенты при свободном члене и $\boldsymbol{\beta}$ соответственно:

$$\mathrm{c}_0 = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha, \quad \mathrm{c}_1 = \mathrm{a}_0\mathrm{b}_1 + \mathrm{a}_1\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1.$$

Коэффициент c_0 вычисляется по формуле $\mathrm{c}_0 = \mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha$, где

— $\mathrm{a}_0\mathrm{b}_0$ и $\mathrm{a}_1\mathrm{b}_1$ — операции умножения в подполе $\mathbb{F}_{2^2}$;
— $(\mathrm{a}_1\mathrm{b}_1)\,\alpha$ — операция умножения на константу α в подполе $\mathbb{F}_{2^2}$;
— $(\mathrm{a}_0\mathrm{b}_0) + (\mathrm{a}_1\mathrm{b}_1\alpha)$ — операция сложения в подполе $\mathbb{F}_{2^2}$.

Тогда, учитывая, что:

— умножение в $\mathbb{F}_{2^2}$ имеет комбинационную сложность $C_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 7$ и глубину $D_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) \leqslant 3$ (см. предложение 1);
— умножение на константу α в $\mathbb{F}_{2^2}$ имеет комбинационную сложность $C_\Omega(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$ и глубину $D_\Omega(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$ (см. предложение 2);
— сложение в $\mathbb{F}_{2^2}$ имеет сложность $C_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) = 2$ и глубину $D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) = 1$,

получаем, что для c_0:

$$C_\Omega\left(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2 \cdot 7 + 1 + 2 = 17,$$

$$D_\Omega\left(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant \max\Big\{ D_\Omega\left(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}\right) + D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}),$$

$$D_\Omega\left(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}\right) + D_\Omega\left(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}\right) + D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) \Big\} \leqslant \max\{4, 5\} = 5.$$

Коэффициент c_1 после упрощений может быть вычислен по формуле

$$\mathrm{c}_1 = (\mathrm{a}_0 + \mathrm{a}_1)\,(\mathrm{b}_0 + \mathrm{b}_1) + \mathrm{a}_0\mathrm{b}_0,$$

где

— $\mathrm{a}_0 + \mathrm{a}_1$, $\mathrm{b}_0 + \mathrm{b}_1$, $((\mathrm{a}_0 + \mathrm{a}_1)\,(\mathrm{b}_0 + \mathrm{b}_1)) + (\mathrm{a}_0\mathrm{b}_0)$ — операции сложения в подполе $\mathbb{F}_{2^2}$;
— $(\mathrm{a}_0 + \mathrm{a}_1)(\mathrm{b}_0 + \mathrm{b}_1)$ — операция умножения в подполе $\mathbb{F}_{2^2}$;
— $\mathrm{a}_0\mathrm{b}_0$ — операция умножения в подполе $\mathbb{F}_{2^2}$, уже выполненная при вычислении коэффициента c_0.

Таким образом, для c_1:

$$C_\Omega\big(\mathrm{c}_1; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) \leqslant$$

$$\leqslant 2\,C_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) + C_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}) + C_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) = 13,$$

$$D_\Omega\big(\mathrm{c}_1; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) \leqslant$$

$$\leqslant \max\Big\{ D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) + D_\Omega(x \cdot \alpha; \mathbb{F}_{2^2}; \mathrm{Norm}) + D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}),$$

$$D_\Omega(x \cdot y; \mathbb{F}_{2^2}; \mathrm{Norm}), D_\Omega(x + y; \mathbb{F}_{2^2}; \mathrm{Norm}) \Big\} \leqslant 5.$$

Операция умножения $\mathbf{a} \cdot \mathbf{b}$ требует вычисления c_0 и c_1. Сложность и глубина оцениваются следующим образом:

$$C_\Omega\big(\mathbf{a} \cdot \mathbf{b}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) = C_\Omega(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}) + C_\Omega(\mathrm{c}_1; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}) \leqslant 17 + 13 = 30,$$

$$D_\Omega\big(\mathbf{a} \cdot \mathbf{b}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\,\big) = \max(D_\Omega(\mathrm{c}_0; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}), D_\Omega(\mathrm{c}_1); \mathbb{F}_{(2^2)^2}; \mathrm{Poly})) \leqslant \max(5, 5) = 5.$$

Предложение 4 доказано. ■

Заметим, что доказательство предложения 4 является чисто техническим, для него достаточно рассмотреть следующую реализацию операции умножения:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})\,(\mathrm{b}_0 + \mathrm{b}_1\boldsymbol{\beta}) = \\ = (\mathrm{a}_0\mathrm{b}_0 + \mathrm{a}_1\mathrm{b}_1\alpha) + ((\mathrm{a}_0 + \mathrm{a}_1)\,(\mathrm{b}_0 + \mathrm{b}_1) + \mathrm{a}_0\mathrm{b}_0)\,\boldsymbol{\beta} = \mathrm{c}_0 + \mathrm{c}_1\boldsymbol{\beta} = \mathbf{c}.$$

Предложение 5. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется $C_\Omega\left(x^4; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2$; $D_\Omega\left(x^4; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) = 1$.

Действительно, возведение в степень 4 является эндоморфизмом Фробениуса и вычисляется по формуле

$$\mathbf{a}^4 = (\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})^4 = \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}^4 = \mathrm{a}_0 + \mathrm{a}_1(\boldsymbol{\beta} + \alpha^2 + \alpha) = \mathrm{a}_0 + \mathrm{a}_1(\boldsymbol{\beta} + 1) = (\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}.$$

Возведение в квадрат вычислется аналогично, откуда следует

Предложение 6. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x^2; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 3$; $D_\Omega\left(x^2; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2$.

Обратный элемент в поле можно вычислить с использованием алгоритма Ито — Цудзи [34], в котором используется следующее свойство: для любого ненулевого элемента $\mathbf{a} \in \mathbb{F}_{(2^2)^2}$ выполняется $\mathbf{a}^5 \in \mathbb{F}_{2^2}$. Это следует из того, что мультипликативная группа $\mathbb{F}_{2^2}^\times$ имеет порядок 3, а, поскольку $\mathbf{a}^{15} = 1$, то $(\mathbf{a}^5)^3 = 1$, что означает $\mathbf{a}^5 \in \mathbb{F}_{2^2}$. Таким образом, реализация операции обращения в $\mathbb{F}_{(2^2)^2}$ сводится к реализации операции обращения в подполе:

$$\mathbf{a}^{-1} = \left(\mathbf{a}\mathbf{a}^4\right)^{-1}\mathbf{a}^4 = \left((\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta})\left(\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}^4\right)\right)^{-1}\left(\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}^4\right) = \\ = \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}) = \mathrm{d}_0 + \mathrm{d}_1\boldsymbol{\beta}.$$

Вычисляя выражения для значений коэффициентов d_0 и d_1, получаем

Предложение 7. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется $C_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 26$; $D_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8$.

Так как в рамках данной работы мы ограничиваемся выбором только мономиальных параметров в параметрических семействах типов «А» и «Б» для построения подстановок пространства $\mathbb{F}_2^8$, то найдём сложности реализации всех подстановок вида x^i, $i = 1, \ldots, 15$. Все такие подстановки разбиваются на два класса: линейные при $i \in \{1, 2, 4, 8\}$ и нелинейные при $i \in \{7, 11, 13, 14\}$.

Сначала рассмотрим подстановку x^8.

Предложение 8. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x^8; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 3$; $D_\Omega\left(x^8; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 2$.

Доказательство следует из следующей цепочки равенств:

$$\mathbf{a}^8 = (\mathbf{a}^4)^2 = ((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta})^2 = (\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2\boldsymbol{\beta}^2 = (\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2(\boldsymbol{\beta} + \alpha) = (\mathrm{a}_0^2 + \mathrm{a}_1^2\alpha^2) + \mathrm{a}_1^2\boldsymbol{\beta}.$$

Цикломатический класс элемента α в мультипликативной группе конечного поля $\mathbb{F}_{2^n}$ образуют все элементы вида α^{2^k} для $k \geqslant 0$, соответствующие сопряжённым элементам относительно эндоморфизма Фробениуса $\phi(x) = x^2$. Для подстановок в поле $\mathbb{F}_{2^4}$ выполняются следующие соотношения внутри одного цикломатического класса:

$$x^{14} = x^{-1}, \quad x^{13} = (x^{-1})^2 = x^{-2}, \quad x^{11} = (x^{-1})^4 = x^{-4}, \quad x^7 = (x^{-1})^8 = x^{-8}.$$

Данные равенства следуют из того, что в мультипликативной группе поля $\mathbb{F}_{2^4}^{\times}$ порядка 15 выполняется $x^{15} = 1$ для всех $x \neq 0$, следовательно $x^{-k} \equiv x^{15-k}$, где $1 \leqslant k < 15$.

Предложение 9. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется

$$C_\Omega\left(x^{13}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 29, \quad D_\Omega\left(x^{13}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8;$$
$$C_\Omega\left(x^{11}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 26, \quad D_\Omega\left(x^{11}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8;$$
$$C_\Omega\left(x^{7}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 29, \quad D_\Omega\left(x^{7}; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 8.$$

Доказательство. Используя соотношение $x^{11} = (x^{-1})^4$ и предложения 3, 5 и 7, получаем

$$\mathbf{a}^{11} = (\mathbf{a}^{-1})^4 = \left[\left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}\right)\right]^4 =$$
$$= \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-4}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}^4\right) = \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta}^4\right) =$$
$$= \left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}(\mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta}).$$

Действительно:

— элемент $(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha) = \mathbf{a}^5$ принадлежит подполю $\mathbb{F}_{2^2}$ (предложение 7), мультипликативная группа которого имеет порядок 3;

— $\boldsymbol{\beta}^4 = \boldsymbol{\beta} + 1$ (предложение 5).

Для x^{13} и x^7 доказательство аналогично с использованием соотношений $x^{13} = (x^{-1})^2$ и $x^7 = (x^{-1})^8$, при этом глубина вычислений не превышает установленных границ благодаря мультипликативной структуре подполя $\mathbb{F}_{2^2}$. В качестве примера рассмотрим x^{13}:

$$\mathbf{a}^{13} = \left(\mathbf{a}^{14}\right)^2 = \left[\left(\mathrm{a}_0(\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1^2\alpha\right)^{-1}\left((\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\right)\boldsymbol{\beta}\right]^2 =$$
$$= (\mathrm{a}_0^2(\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1\alpha^2)^{-1}((\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2\boldsymbol{\beta}^2) = (\mathrm{a}_0^2(\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1\alpha^2)^{-1}((\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1^2(\boldsymbol{\beta} + \alpha)) =$$
$$= \left(\mathrm{a}_0^2(\mathrm{a}_0^2 + \mathrm{a}_1^2) + \mathrm{a}_1\alpha^2\right)^{-1}\left(\mathrm{a}_0^2 + \mathrm{a}_1^2\alpha^2 + \mathrm{a}_1^2\boldsymbol{\beta}\right).$$

Предложение 9 доказано. ■

Как указано ранее (замечание 3), нас также интересует функция $x^7 + x^4 + x$. Рассмотрим сначала функцию $x^4 + x$:

$$\mathbf{a}^4 + \mathbf{a} = (\mathrm{a}_0 + \mathrm{a}_1) + \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0 + \mathrm{a}_1\boldsymbol{\beta} = \mathrm{a}_1.$$

Тогда очевидно

Предложение 10. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в полиномиальном базисе $\{1, \boldsymbol{\beta}\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ выполняется $C_\Omega\left(x^7 + x^4 + x; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 31$; $D_\Omega\left(x^7 + x^4 + x; \mathbb{F}_{(2^2)^2}; \mathrm{Poly}\right) \leqslant 9$.

4.3. Сложность задания функций над полем $\mathbb{F}_{(2^2)^2}$ в нормальном и смешанном базисах

Пусть элементы поля $\mathbb{F}_{(2^2)^2}$ задаются в нормальном базисе. Известно, что с использованием нормального базиса эффективно реализуется эндоморфизм Фробениуса. Это позволяет предложить, что реализация операции обращения ненулевых элементов x^{-1} имеет меньшую сложность.

Как и в работе [8], рассмотрим нормальный базис $\{\boldsymbol{\beta}, \boldsymbol{\beta}^4\}$, где $\boldsymbol{\beta}$ — корень неприводимого над $\mathbb{F}_{2^2}$ многочлена $x^2+x+\alpha$. Здесь, как и выше, α — образующий нормального базиса $\mathbb{F}_{2^2}$. Пусть $\mathbf{a} = \mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4$ — произвольный ненулевой элемент поля $\mathbb{F}_{(2^2)^2}$. Обратный элемент в поле вычисляется по формуле

$$\mathbf{a}^{-1} = \left(\mathbf{a}\mathbf{a}^4\right)^{-1}\mathbf{a}^4 = \left(\left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right)\right)^{-1}\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right) =$$
$$= \left(\mathrm{a}_0\mathrm{a}_1 + (\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right) = \mathrm{d}_0\boldsymbol{\beta} + \mathrm{d}_1\boldsymbol{\beta}^4.$$

Записав явные выражения для d_0 и d_1, получаем

Предложение 11. Пусть поле $\mathbb{F}_{(2^2)^2}$ задаётся неприводимым многочленом $g(x) = x^2 + x + \alpha$ в нормальном базисе $\{\boldsymbol{\beta}, \boldsymbol{\beta}^4\}$. Тогда для $x \in \mathbb{F}_{(2^2)^2}$ имеет место $C_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Norm}\right) \leqslant 26$; $D_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{Norm}\right) \leqslant 7$.

Таким образом, использование нормального базиса позволяет сократить глубину функции, реализующей вычисление обратного элемента в поле, на 1.

В работе [18] предлагается рассматривать смешанные базисы для реализации операций в поле. Использование разных базисов для разных операций может привести к уменьшению глубины схемы, реализующей подстановку в целом. Например, следующая формула описывает способ реализации подстановки обращения ненулевых элементов, заданных в нормальном базисе, так, что результат представлен в полиномиальном базисе:

$$\mathbf{a}^{-1} = \left(\mathbf{a}\mathbf{a}^4\right)^{-1}\mathbf{a}^4 = \left(\left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right)\right)^{-1}\left(\mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4\right) =$$
$$= \left(\mathrm{a}_0\mathrm{a}_1 + (\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left((\mathrm{a}_1+\mathrm{a}_0) + \mathrm{a}_0\boldsymbol{\beta}\right) = \mathrm{d}_0 + \mathrm{d}_1\boldsymbol{\beta}.$$

Здесь d_0 и d_1 — коэффициенты в полниномиальном базисе. Такое представление не приводит к увеличению сложности формулы. Для смешанных базисов будем использовать обозначения PtN (Polynomialal to Normal) и NtP (Normal to Polynomial) для функций, задаваемых в одном базисе, результат которых представляется в другом базисе. В качестве полиномиального базиса везде далее будем рассматривать $\{1, \boldsymbol{\beta}\}$, а в качестве нормального — базис $\{\boldsymbol{\beta}, \boldsymbol{\beta}^4\}$.

Предложение 12 [18]. Для $x \in \mathbb{F}_{(2^2)^2}$ выполнено $C_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{NtP}\right) \leqslant 26$; $D_\Omega\left(x^{-1}; \mathbb{F}_{(2^2)^2}; \mathrm{NtP}\right) \leqslant 7$.

Найдём сложность реализации всех мономиальных подстановок. Возведение в степень 4 вычисляется по формуле

$$\mathbf{a}^4 = \left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)^4 = \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4.$$

Если необходимо, чтобы результат был представлен в полиномиальном базисе, то получаем

$$\mathbf{a}^4 = \left(\mathrm{a}_0\boldsymbol{\beta} + \mathrm{a}_1\boldsymbol{\beta}^4\right)^4 = \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0\boldsymbol{\beta}^4 = \mathrm{a}_1\boldsymbol{\beta} + \mathrm{a}_0(\boldsymbol{\beta}+1) = \mathrm{a}_0 + (\mathrm{a}_0+\mathrm{a}_1)\boldsymbol{\beta}.$$

Отсюда верно

Предложение 13. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=0,\quad D_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=0,$$
$$C_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 2,\quad D_\Omega\left(x^4;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 1.$$

Возведение в квадрат и степень 8 вычисляются аналогично:

Предложение 14. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=C_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)\leqslant 4,$$
$$D_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)=D_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right)\leqslant 2.$$

Доказательство. Для элемента $\mathbf{a}=\mathrm{a}_0\boldsymbol{\beta}+\mathrm{a}_1\boldsymbol{\beta}^4\in\mathbb{F}_{(2^2)^2}$, используя равенства $\boldsymbol{\beta}^2=\boldsymbol{\beta}+\alpha$ и $\boldsymbol{\beta}^8=\boldsymbol{\beta}^4+\alpha$, получаем

$$\mathbf{a}^2=(\mathrm{a}_0\boldsymbol{\beta}+\mathrm{a}_1\boldsymbol{\beta}^4)^2=\mathrm{a}_0^2\boldsymbol{\beta}^2+\mathrm{a}_1^2\boldsymbol{\beta}^8=\mathrm{a}_0^2(\boldsymbol{\beta}+\alpha)+\mathrm{a}_1^2(\boldsymbol{\beta}^4+\alpha)=(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha)\boldsymbol{\beta}+(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha^2)\boldsymbol{\beta}^4.$$

Аналогично $\mathbf{a}^8=(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)\boldsymbol{\beta}+(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)\boldsymbol{\beta}^4$.

Для завершения доказательства заметим, что вычисление $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ можно произвести за три (а не четыре) операции, так как значение этой функции зависит от пересекающихся значений переменных. После этого значение $(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)$ вычисляется умножением $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ на α^2, которое возможно произвести за одну операцию.

Для получения оценки глубины необходимо рассмотреть граф, в котором $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ и $(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)$ — это два независимых пути. ■

Аналогично для смешанного базиса:

Предложение 15. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)=C_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 4,$$
$$D_\Omega\left(x^2;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)=D_\Omega\left(x^8;\mathbb{F}_{(2^2)^2};\mathrm{NtP}\right)\leqslant 2.$$

Таким образом, использование нормального базиса для реализации линейных подстановок не повышает эффективности по сравнению с использованием полиномиального базиса.

Оценим сложность реализации подстановок x^7, x^{11}, x^{13}. Для получения итоговой формулы, как и ранее, необходимо явно получить выражения для коэффициентов d_0 и d_1. Рассмотрим сначала самый простой случай:

$$\mathbf{a}^{11}=\left(\mathbf{a}^{-1}\right)^4=\left(\mathrm{a}_0\mathrm{a}_1+(\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left(\mathrm{a}_0\boldsymbol{\beta}+\mathrm{a}_1\boldsymbol{\beta}^4\right)=\mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Если аргумент подстановки представляется в полиномиальном базисе, то получаем

$$\mathbf{a}^{11}=\left(\mathbf{a}^{-1}\right)^4=\left(\mathrm{a}_0\mathrm{a}_1+(\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}\left((\mathrm{a}_0+\mathrm{a}_1)+\mathrm{a}_0(\boldsymbol{\beta}+1)\right)=$$
$$=\left(\mathrm{a}_0\mathrm{a}_1+(\mathrm{a}_0+\mathrm{a}_1)^2\alpha\right)^{-1}(\mathrm{a}_1+\mathrm{a}_0\boldsymbol{\beta})=\mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Замечание 5. Комбинационная сложность и глубина функции, реализующей подстановку x^{11} в нормальном и смешанном базисах, равны соответствующим значениям для подстановки x^{14}.

Рассмотрим подстановку x^{13} в нормальном и смешанном базисах:

$$\mathbf{a}^{13} = \left(\mathbf{a}^{-1}\right)^2 = \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\mathrm{a}_1^2\left(\alpha^2\boldsymbol{\beta}+\alpha\boldsymbol{\beta}^4\right)+\mathrm{a}_0^2\left(\alpha\boldsymbol{\beta}+\alpha^2\boldsymbol{\beta}^4\right)\right) =$$
$$= \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\left(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2\right)\boldsymbol{\beta}+\left(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha\right)\boldsymbol{\beta}^4\right) = \mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Для подстановки $x^7 = (x^{13})^4$ формула останется такой же с точностью до перестановки значений коэффициентов при базисных векторах.

Предложение 16. Для $x \in \mathbb{F}_{(2^2)^2}$

$$C_\Omega\left(x^{13};\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) = C_\Omega\left(x^7;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) \leqslant 29,$$
$$D_\Omega\left(x^{13};\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) = D_\Omega\left(x^7;\mathbb{F}_{(2^2)^2};\mathrm{Norm}\right) \leqslant 7.$$

Доказательство. Достаточно показать, что значения $(\mathrm{a}_0+\mathrm{a}_1)\alpha^2$, $(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2)$, $(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha)$ можно вычислить за шесть операций в базисе Ω. ■

Рассмотрим случай, когда значение подстановки представляется в полиномиальном базисе:

$$\mathbf{a}^{13} = \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\left(\mathrm{a}_0^2\alpha+\mathrm{a}_1^2\alpha^2\right)\boldsymbol{\beta}+\left(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha\right)(\boldsymbol{\beta}+1)\right) =$$
$$= \left(\mathrm{a}_0^2\mathrm{a}_1^2 + (\mathrm{a}_0+\mathrm{a}_1)\alpha^2\right)^{-1}\left(\left(\mathrm{a}_0^2\alpha^2+\mathrm{a}_1^2\alpha\right)+\left(\mathrm{a}_0^2+\mathrm{a}_1^2\right)\boldsymbol{\beta}\right) = \mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Замечание 6. Комбинационная сложность и глубина функций, реализующих подстановки x^{13} и x^7 в нормальном базисе, равны соответствующим значениям в смешанном базисе.

Приведём формулу из [18], позволяющую получить значение в нормальном базисе для операции умножения в поле элементов, представленных в полиномиальном базисе:

$$\mathbf{a}\cdot\mathbf{b} = (\mathrm{a}_0+\mathrm{a}_1\boldsymbol{\beta})(\mathrm{b}_0+\mathrm{b}_1\boldsymbol{\beta}) = \left[(\mathrm{a}_0+\mathrm{a}_1)(\mathrm{b}_0+\mathrm{b}_1)+\mathrm{a}_1\mathrm{b}_1\alpha\right]\boldsymbol{\beta}+(\mathrm{a}_0\mathrm{b}_0+\mathrm{a}_1\mathrm{b}_1\alpha)\boldsymbol{\beta}^4 = \mathrm{d}_0+\mathrm{d}_1\boldsymbol{\beta}.$$

Предложение 17. Для $x, y \in \mathbb{F}_{(2^2)^2}$ выполнено

$$C_\Omega\left(x\cdot y;\mathbb{F}_{(2^2)^2};\mathrm{PtN}\right) \leqslant 26; \quad D_\Omega\left(x\cdot y;\mathbb{F}_{(2^2)^2};\mathrm{PtN}\right) \leqslant 5.$$

Такое представление имеет комбинационную сложность на 4 меньше, чем в случае полиномиального базиса, однако обе реализации имеют равную глубину.

4.4. Сложность задания функций над полем $\mathbb{F}_{(2^2)^2}$ с использованием PRR- и RRB-представлений

Пусть $f(x) = x^4+x^3+x^2+x+1$ — неприводимый многочлен над полем $\mathbb{F}_2$ и $\boldsymbol{\beta}$ — его корень в минимальном поле разложения. Тогда $\{\boldsymbol{\beta}^0,\boldsymbol{\beta}^1,\boldsymbol{\beta}^2,\boldsymbol{\beta}^3\}$ — полиномиальный базис. При использовании RRB-представления элементы поля представляются в виде линейной комбинации элементов множества $\{\boldsymbol{\beta}^0,\boldsymbol{\beta}^1,\boldsymbol{\beta}^2,\boldsymbol{\beta}^3,\boldsymbol{\beta}^4\}$. Очевидно, что элементы поля представляются не единственным образом.

Для такого представления операция умножения реализуется следующим образом: пусть $\mathbf{a} = a_0+a_1\boldsymbol{\beta}+\ldots+a_4\boldsymbol{\beta}^4$, $\mathbf{b} = b_0+b_1\boldsymbol{\beta}+\ldots+b_4\boldsymbol{\beta}^4$. Тогда $\mathbf{d} = \mathbf{a}\cdot\mathbf{b}$, $\boldsymbol{d} = d_0+{}$ $+d_1\boldsymbol{\beta}+\ldots+d_4\boldsymbol{\beta}^4$ вычисляются по формулам [18]

$$d_0 = (a_1+a_3)(b_1+b_4)+(a_2+a_3)(b_2+b_3),$$
$$d_1 = (a_0+a_1)(b_0+b_1)+(a_2+a_4)(b_2+b_4),$$
$$d_2 = (a_0+a_2)(b_0+b_2)+(a_4+a_4)(b_3+b_4),$$
$$d_3 = (a_0+a_3)(b_0+b_3)+(a_2+a_2)(b_1+b_2),$$
$$d_4 = (a_0+a_4)(b_0+b_4)+(a_3+a_3)(b_1+b_3).$$

Комбинационная сложность такого представления равна 35, что больше аналогичных значений для других базисов. В то же время глубина функции, задающей такое представление, равна 3, что является наименьшим известным значением [18].

Использование PRR-представления позволяет реализовывать некоторые операции эффективнее, чем в полиномиальном, нормальном или смешанных базисах. Например, в работе [18] приводится способ вычисления обратного элемента в поле.

Пусть $\mathbf{a} = a_0 + a_1\boldsymbol{\beta} + \ldots + a_4\boldsymbol{\beta}^4$, $\mathbf{b} = b_0 + b_1\boldsymbol{\beta} + \ldots + b_4\boldsymbol{\beta}^4$, $\mathbf{a}^{-1} = \mathbf{b}$. Тогда

$$\begin{aligned} b_0 &= (a_1 \vee a_4)(a_2 \vee a_3), \\ b_1 &= ((a_4 + 1)(a_1 + a_2)) \vee (a_0 a_4 (a_2 \vee a_3)), \\ b_2 &= ((a_3 + 1)(a_2 + a_4)) \vee (a_0 a_3 (a_1 \vee a_4)), \\ b_3 &= ((a_2 + 1)(a_1 + a_3)) \vee (a_0 a_2 (a_1 \vee a_4)), \\ b_4 &= ((a_1 + 1)(a_3 + a_4)) \vee (a_0 a_1 (a_2 \vee a_3)). \end{aligned}$$

Такое представление возможно, так как в PRR-базисе аргументами булевой функции, задающей операцию обращения ненулевых элементов поля, являются только векторы, имеющие нулевой двоичный вес. Это позволяет определять остальные значения произвольным образом так, чтобы минимизировать сложность вычислений. Более того, так как значение, заданное в PRR-представлении, также является RRB-представлением, то (как и сделано авторами [18] для функции выше) можно отказаться от требования, что значение подстановки будет иметь PRR-представление. При использовании PRR-представления с $p(x) = (x + 1)(x^4 + x^3 + x^2 + x + 1)$ элемент $1 \in \mathbb{F}_{2^4}$ может быть представлен так:

— как многочлен $g(x) = x^4 + x^3 + x^2 + x + 1$ (каноническое PRR-представление);
— как константа 1 (вырожденное представление).

При последующих умножениях в поле оба представления ведут себя идентично относительно представлений элементов поля, что сохраняет корректность вычислений.

Комбинационная сложность операции обращения ненулевых элементов поля равна 31, что больше, чем в случае нормального, полиномиального и смешанного базисов, однако глубина функции равна 3.

Заметим, что возведение элемента в степени 2, 4, 8 имеет комбинационную сложность и глубину, равные нулю. Отсюда, в частности, следует, что вычисление значения мономиальной подстановки имеет сложность и глубину, равную аналогичным значениям операции вычисления подстановки x^{14}.

4.5. Сложность реализации мультиплексора MUX

Рассмотрим трудоёмкость реализации мультиплексора, аналогично [6]. Согласно определению параметрических семейств типов «А», «Б» и «Г», происходят вычисления, аналогичные следующему:

«Если $x_1 = 0$, то $y = \widehat{\pi}(x_0)$, иначе $y = \pi_2(\pi_0(x_0) \cdot \pi_1(x_1))$»,

где $\pi_0, \pi_1, \pi_2, \widehat{\pi}$ — нелинейные биективные преобразования пространства $\mathbb{F}_2^4$.

Рассмотрим функцию-индикатор, принимающую значение 1 в точке $x_1 = 0$ и нулевое значение во всех остальных точках:

$$\mathrm{Ind}_0(x_1) = \overline{x_1^{(1)}} \cdot \overline{x_1^{(2)}} \cdot \overline{x_1^{(3)}} \cdot \overline{x_1^{(4)}} = \overline{x_1^{(1)} \vee x_1^{(2)} \vee x_1^{(3)} \vee x_1^{(4)}}.$$

Здесь $x_1^{(j)}$, $j = 1, \ldots, 4$, — значение j-й координаты вектора $x_1 \in \mathbb{F}_2^4$. Комбинационная сложность вычисления функции индикатора равна 4, глубина равна 3 (за счёт вычисления операции отрицания).

Значение рассматриваемой функции может быть вычислено по формуле

$$\mathrm{Ind}_0(x_1) \cdot \widehat{\pi}(x_0) + \overline{\mathrm{Ind}_0(x_1)} \cdot \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)).$$

Вычисление можно упростить следующим образом:

$$\mathrm{Ind}_0(x_1) \cdot (\widehat{\pi}(x_0) + \pi_2(\pi_0(x_0) \cdot \pi_1(0))) + \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)).$$

Если $\pi_1(0) = 0$, последнее выражение упрощается:

$$\mathrm{Ind}_0(x_1) \cdot (\widehat{\pi}(x_0) + \pi_2(0)) + \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)).$$

Если $\pi_2(0) = 0$, то выражение принимает следующий вид:

$$\mathrm{Ind}_0(x_1) \cdot \widehat{\pi}(x_0) + \pi_2(\pi_0(x_0) \cdot \pi_1(x_1)). \tag{3}$$

Предложение 18. Комбинационная сложность вычисления значения (3) оценивается сверху величиной

$$C_\Omega\left(\widehat{\pi}\right) + C_\Omega\left(\pi_2(\pi_0(x_0) \cdot \pi_1(x_1))\right) + 12.$$

Глубина функции, реализующей (3), равна

$$\max\left\{4, D_\Omega\left(\widehat{\pi}\right) + 2, D_\Omega\left(\pi_2(\pi_0(x_0) \cdot \pi_1(x_1))\right) + 1\right\}.$$

Замечание 7. При использовании PRR-представления сложность и глубина формулы, вычисляющей $\mathrm{Ind}_0(x_1)$, не изменится, так как нулевое значение в этом представлении задается вектором из пяти нулей [18], при этом ни один из других векторов $\mathbb{F}_2^5$, задающих элементы поля $\mathbb{F}_2^4$, не имеет в своей записи четыре нуля ни на каких позициях.

В случае RRB-представления комбинационная сложность вычисления $\mathrm{Ind}_0(x_1)$ увеличится на 1, а глубина не изменится. Более того, для подстановок, сохраняющих 0 (мономиальные подстановки сохраняют 0), можно вычислять $\mathrm{Ind}_0(x_1)$ от входных значений, что даже при использовании PRR-представления не изменит комбинационную сложность (относительно значения, полученного в предложении 18) и позволит сократить глубину всей схемы целиком.

Таким образом, для нормального и полиномиального базисов комбинационная сложность вычисления y на 12 больше сложности вычисления оставшихся функций; для PRR- или RRB-представлений она равна 14.

4.6. Сложность реализации $\widehat{\pi}_i$

Как сказано ранее, в результате экспериментальных исследований алгоритма из работы [27] выяснилось, что подстановки $\widehat{\pi}_i$, $i \in \{1, 2\}$, в подавляющем большинстве аффинно эквивалентны подстановкам с представителями x^{14}, $x^7 + x^4 + x$, сложность реализации которых уже оценена.

При реализации аффинно эквивалентных подстановок помимо формул x^{14} и $x^7 + x^4 + x$ необходимо вычислить не более двух умножений на обратимые матрицы из $\mathrm{GL}(4, 2)$ и не более двух сложений с векторами длины 4.

Глубина умножения на матрицу зависит от максимального веса $\mathrm{w}_{\max}$ строки матрицы и равна $\lceil \log_2 \mathrm{w}_{\max} \rceil$. Это значение всегда меньше либо равно 2.

Комбинационную сложность можно оценить с использованием количества единиц во всей матрице. Можно легко показать, что комбинационная сложность умножения

на произвольную (но фиксированную) обратимую матрицу не превосходит 9. Действительно, при умножении вектора из $\mathbb{F}_2^4$ на произвольную обратимую (4×4)-матрицу максимальная комбинационная сложность не превышает девяти операций «$\oplus$». Это достигается за счёт оптимального выбора предвычисляемых парных сумм входных переменных.

Для четырёх переменных существует $\mathrm{C}_4^2 = 6$ возможных попарных сумм, однако в любом случае достаточно вычислить только пять из них. Выбор конкретных пар определяется структурой матрицы. Такое предвычисление требует ровно пять операций «$\oplus$».

Для строк, содержащих менее трёх единиц, дополнительных вычислений не потребуется. При вычислении выходных значений для строк матрицы, содержащих три единицы, понадобится одна дополнительная операция «$\oplus$», если использовать предвычисленные пары. Для строк с четырьмя единицами выходное значение получается суммированием двух предвычисленных пар, что добавляет ещё одну операцию.

В наиболее требовательном случае, когда матрица содержит одну строку с четырьмя единицами и три строки с тремя единицами, общее количество операций составляет

$$5 \text{ (предвычисление)} + 1 \text{ (4 единицы)} + 3 \times 1 \text{ (3 единицы)} = 9 \text{ операций «}\oplus\text{».}$$

Для конкретных матриц сложность может быть ниже, но представленный метод гарантирует, что в любом случае девяти операций «$\oplus$» достаточно для выполнения матричного умножения над $\mathbb{F}_2^4$.

Таким образом, комбинационная сложность вычислений $\widehat{\pi}_i$, $i \in \{1, 2\}$, не превосходит $C_\Omega(\pi')+26$; глубина формулы не превосходит $D_\Omega(\pi')+6$, где $\pi' \in \{x^{14}, x^7 + x^4 + x\}$.

Замечание 8. Для заданной подстановки $\widehat{\pi}_i$ могут существовать различные аффинные представления вида

$$\widehat{\pi}_i(x) = \mathcal{A}_1 \circ \pi' \circ \mathcal{A}_2(x) = \mathcal{B}_1 \circ \pi' \circ \mathcal{B}_2(x),$$

где $\pi' \in \{x^{-1}, x^7+x^4+x\}$; $\mathcal{A}_k, \mathcal{B}_k$, $k = 1, 2,$ — аффинные преобразования. Это позволяет выбирать между представлениями, оптимизированными по разным критериям: одно может минимизировать комбинационную сложность, другое — глубину схемы.

4.7. Сложность реализации подстановок из рассматриваемых параметрических семейств

Рассмотрим сложность реализации подстановок из параметрического семейства типа «Г», которые в том числе обобщают параметрические подстановки типов «А» и «Б» в случае мономиального выбора параметров π_1, π_2.

Будем считать, что входные векторы заданы в нужном базисе, так как умножение каждой из координат x_1, x_2 на обратимую матрицу не изменяет её класс эквивалентности, при этом представление операций при их задании не конкретизируется.

Проведём рассуждения аналогично работе [33]. Для этого вычисление всех подстановок и операции умножения необходимо проводить в смешанных базисах, а вычисление $\widehat{\pi}_i$, $i = 1, 2,$ — в нормальном базисе. Для подстановки из параметрического семейства типа «Г» необходимо реализовать две функции, каждая из которых состоит из трёх подстановок (две из которых мономиальные), операции умножения и мультиплексора. Комбинационная сложность задания такой подстановки оценивается следующей величиной:

$$C_\Omega(x^\alpha) + C_\Omega(x^\beta) + C_\Omega(x^\gamma) + C_\Omega(x^\delta) + 2C_\Omega(\cdot) + C_\Omega\left(\widehat{\pi}_1\right) + C_\Omega\left(\widehat{\pi}_2\right) + 2C_\Omega(\mathrm{MUX}) \leqslant$$
$$\leqslant 4 \cdot C_\Omega(x^7) + 2C_\Omega(\cdot) + 2C_\Omega(x^7 + x^4 + x) + 2 \cdot 26 + 2 \cdot 12 = 314.$$

Более того, каждая из пар подстановок (x^α, x^γ), $\left(x^\beta, x^\delta\right)$ либо содержит одну линейную подстановку и реализация пары не превышает $29+3$ операций, либо содержит две нелинейные подстановки, формулы вычисления которых во многом совпадают и их комбинационная сложность также не превосходит $29+3$ (сначала вычисляем нелинейную, затем возводим в степень 2^i при некотором i и опять получаем другую нелинейную). Этот факт позволяет уменьшить максимальное значение комбинационной сложности до 262.

Эта величина может быть сильно завышенной. Рассмотрим следующую подстановку $S(x_1, x_2) = (y_1, y_2)$ [22, 23]:

$$\begin{aligned} y_1 &= \begin{cases} x_1 \cdot x_2^2, & x_2 \neq 0, \\ x_1^{-1}, & x_2 = 0, \end{cases} \\ y_2 &= \begin{cases} x_1^{-1} \cdot x_2^{-1}, & x_1 \neq 0, \\ x_2^{-1}, & x_1 = 0. \end{cases} \end{aligned} \tag{4}$$

Её комбинационная сложность в рассматриваемом базисе не превосходит 139.

Оценим глубину формулы, задающей подстановку из параметрического семейства типа «Г»:

$$\begin{gathered} \max\left\{4, D_\Omega\left(\widehat{\pi}_1\right)+2, D_\Omega\left(\widehat{\pi}_2\right)+2, D_\Omega\left(x_1^\alpha \cdot x_2^\beta\right)+1, D_\Omega\left(x_1^\gamma \cdot x_2^\delta\right)+1\right\} \leqslant \\ \leqslant \max\left\{4, 8+8+2, \max\left\{D_\Omega\left(x_1^\alpha\right), D_\Omega\left(x_2^\beta\right), D_\Omega\left(x_1^\gamma\right), D_\Omega\left(x_2^\delta\right)\right\}+6\right\} \leqslant \\ \leqslant \max\{4, 8+8+2, 8+6\} \leqslant 18. \end{gathered}$$

Глубина формулы, задающей подстановку (4), очевидно, не превышает 14. Из этого, в частности, следует, что для реализации подстановок на программно-аппаратных платформах существенным является выбор именно $\widehat{\pi}_i$, $i = \{1, 2\}$. При той же глубине подстановка $G_\Gamma(x_1, x_2)$, определённая на с. 35, имеет комбинационную сложность 136.

В случае, когда все $i \in \{\alpha, \beta, \gamma, \delta\}$, задающие мономиальные подстановки, принадлежат множеству $\{4, 7, 11, 13, 14\}$, можно использовать представление подстановок в нормальном базисе для уменьшения глубины соответствующей цепочки на 1. При этом комбинационная сложность реализации конкретных подстановок может не измениться $(1, 7, 11, 13, 14)$, уменьшиться (4) или увеличиться $(2, 8)$. Использование смешанных базисов позволяет сократить комбинационную сложность операции умножения. Например, комбинационная сложность подстановки S (с использованием смешанных базисов) в случае задания аргументов в нормальном базисе равна 134 (две дополнительные операции на вычисление единичной подстановки в смешанном базисе). Глубина формулы, задающей подстановку, равна 13. При той же глубине подстановка $G_\Gamma(x_1, x_2)$ имеет комбинационную сложность 130.

Получается, что применение смешанных базисов потенциально позволяет снизить как комбинационную сложность, так и глубину формулы, задающей подстановку.

Использование PRR- и RRB-представлений оправдано только при необходимости минимизации глубины формулы, задающей подстановку. Однако могут они быть полезны при реализации большого количества линейных мономиальных подстановок. Дополнительно необходимо учитывать трудоёмкость преобразования из полиномиального/нормального базисов в эти представления и обратно.

5. Пример получения оценки комбинационной сложности и глубины функции для подстановки пространства $\mathbb{F}_2^8$, используемой в отечественных симметричных алгоритмах

В работе [6] получена оценка комбинационной сложности подстановки π_R, используемой в отечественном алгоритме шифрования с длиной блока 128 бит «Кузнечик» и в отечественном алгоритме хеширования «Стрибог». Как известно [29], эта подстановка имеет TU-представление и для неё применимы результаты, представленные выше. В [6] использован эвристический алгоритм поиска представления преобразований, определяемых TU-представлением, подстановки π_R. Согласно [29], происходят следующие вычисления:

1) $(l \parallel r) := \alpha(l \parallel r)$;
2) если $r = 0$, то $l := \nu_0(l)$, иначе $l := \nu_1(l \cdot I(r))$;
3) $r := \sigma(r \cdot \varphi(l))$;
4) $(l \parallel r) := \omega(l \parallel r)$.

При этом ν_0, ν_1, I, σ — нелинейные биективные функции пространства $\mathbb{F}_2^4$; $\alpha, \omega \in \mathrm{GL}(8,2)$; φ — нелинейное преобразование (подробнее см. в [29]).

В [6] для реализации умножения в поле использован полиномиальный базис $\mathbb{F}_{2^4}$, комбинационная сложность которого составляет 31. Выше показано, что в поле $\mathbb{F}_{(2^2)^2}$ комбинационная сложность умножения равна 30. Воспользуемся этим фактом. Для этого определим преобразования $si, sp \in \mathrm{GL}(4,2)$: si — подстановка, ставящая в соответствие каждому элементу $x \in \mathbb{F}_{2^4}$ его представление в $\mathbb{F}_{(2^2)^2}$; $sp = si^{-1}$ — обратное преобразование. Стоит отметить, что преобразование si (и, как следствие, sp) определены неоднозначно — всего существует 8 таких подстановок, однако следующие рассуждения верны для любой фиксации подстановки si.

Определим вспомогательное преобразование $\nu_0'(l) = \nu_0(l) + \nu_1(0)$ и рассмотрим вычисление подстановки π_R с использованием умножения в полиномиальном базисе $\mathbb{F}_{(2^2)^2}$:

1) $(l \parallel r) := \alpha(l \parallel r)$;
2) $(l \parallel r) := (si(l) \parallel si(r))$;
3) $l := \mathrm{Ind}(r = 0) \cdot \nu_0'(sp(l)) + \nu_1\left(sp\left((l \cdot si(I(sp(r))))\right)\right)$;
4) $r := \sigma\left(sp(r \cdot si(\varphi(l)))\right)$;
5) $(l \parallel r) := \omega(l \parallel r)$.

При этом умножение уже происходит в $\mathbb{F}_{(2^2)^2}$. Преобразования, задаваемые первыми двумя шагами, — некоторое новое линейное преобразование $\widehat{\alpha}$. Введём также следующие обозначения:

— $\widehat{I}(x) = si(I(sp(x)))$;
— $\widehat{\nu_0}(x) = \nu_0'(sp(x))$;
— $\widehat{\nu_1}(x) = \nu_1(sp(x))$;
— $\widehat{\varphi}(x) = si(\varphi(x))$;
— $\widehat{\sigma}(x) = \sigma(sp(x))$.

Тогда алгоритм вычисления подстановки π_R можно переписать так:

1) $(l \parallel r) := \widehat{\alpha}(l \parallel r)$;
2) $l := \mathrm{Ind}(r = 0) \cdot \widehat{\nu_0}(l) + \widehat{\nu_1}(l \cdot \widehat{I}(r))$;
3) $r := \widehat{\sigma}\left(r \cdot \widehat{\varphi}(l)\right)$;
4) $(l \parallel r) := \omega(l \parallel r)$.

Для получения оценок комбинационной сложности и глубины функции, задающей подстановку π_R, необходимо найти соответствующее величины для функций $\widehat{\alpha}$,

$\widehat{I}$, $\widehat{\nu_0}$, $\widehat{\nu_1}$, $\widehat{\varphi}$, $\widehat{\sigma}$. При этом уже корректно говорить, что все преобразования выполняются в полиномиальном базисе (как элементы представляются в памяти компьютера). С использованием эвристического алгоритма [7] построены указанные характеристики сложности (см. приложение):

$$
\begin{aligned}
C_\Omega(\widehat{\alpha};\mathbb{F}_{2^8};\mathrm{Poly}) &= 11, & D_\Omega(\widehat{\alpha};\mathbb{F}_{2^8};\mathrm{Poly}) &= 5;\\
C_\Omega(\widehat{I};\mathbb{F}_{2^4};\mathrm{Poly}) &= 16, & D_\Omega(\widehat{I};\mathbb{F}_{2^4};\mathrm{Poly}) &= 10;\\
C_\Omega(\widehat{\nu_0};\mathbb{F}_{2^4};\mathrm{Poly}) &= 19, & D_\Omega(\widehat{\nu_0};\mathbb{F}_{2^4};\mathrm{Poly}) &= 11;\\
C_\Omega(\widehat{\nu_1};\mathbb{F}_{2^4};\mathrm{Poly}) &= 11, & D_\Omega(\widehat{\nu_1};\mathbb{F}_{2^4};\mathrm{Poly}) &= 7;\\
C_\Omega(\widehat{\varphi};\mathbb{F}_{2^4};\mathrm{Poly}) &= 16, & D_\Omega(\widehat{\varphi};\mathbb{F}_{2^4};\mathrm{Poly}) &= 9;\\
C_\Omega(\widehat{\sigma};\mathbb{F}_{2^4};\mathrm{Poly}) &= 19, & D_\Omega(\widehat{\sigma};\mathbb{F}_{2^4};\mathrm{Poly}) &= 12;\\
C_\Omega(\omega;\mathbb{F}_{2^8};\mathrm{Poly}) &= 5, & D_\Omega(\omega;\mathbb{F}_{2^8};\mathrm{Poly}) &= 2.
\end{aligned}
$$

Отсюда следует, что $C_\Omega(\pi_R;\mathbb{F}_{2^8};\mathrm{Poly}) = 169$, $D_\Omega(\pi_R;\mathbb{F}_{2^8};\mathrm{Poly}) = 55$. Тем же способом можно получить оценку комбинационной сложности и глубины функции, реализующей подстановку $G_\Gamma(x_1, x_2)$, равные $2 \cdot 30 + 2 \cdot 12 + 2 \cdot 16 = 116$ и $7 + 2 + 5 = 14$ соответственно.

Аналогичные результаты, полученные с помощью эвристического алгоритма [7] (однако без перехода в $\mathbb{F}_{(2^2)^2}$), были представлены коллективом авторов на конференции CTCrypt'23 [35], где оценка комбинационной сложности составила 179. Позже авторы улучшили результат и опубликовали новую оценку, равную 176 [36].

Заметим, что полученную оценку можно улучшить следующим образом: пусть $sl, sk \in \mathrm{GL}(4,2)$. Тогда алгоритм вычисления подстановки π_R можно переписать так:

1) $(l \parallel r) := \widehat{\alpha}(l \parallel r)$;
2) $l := \mathrm{Ind}(r = 0) \cdot sl(\widehat{\nu_0})(l) + sl(\widehat{\nu_1})(l \cdot \widehat{I}(r))$;
3) $r := sk(\widehat{\sigma})\,(r \cdot sl^{-1}(\widehat{\varphi})(l))$;
4) $l := sl^{-1}(l)$
5) $r := sk^{-1}(r)$
6) $(l \parallel r) := \omega(l \parallel r)$.

Опробуя значения различных $sl, sk \in \mathrm{GL}(4,2)$, можно попытаться уменьшить оценку комбинационной сложности. Также возможно использовать умножение в смешанном базисе. Всё это является направлением дальнейших исследований.

Выводы

В работе рассмотрены оценки комбинационной сложности и глубины функций, реализующих подстановки из параметрического семейства типа «Г», которые в том числе обобщают параметрические подстановки типов «А» и «Б» в случае мономиального выбора параметров π_1, π_2.

Эти результаты могут быть использованы при реализации указанных подстановок на различных программных и аппаратных платформах. Получены также оценки комбинационной сложности и глубины функции подстановки пространства $\mathbb{F}_2^8$ из отечественных стандартизированных алгоритмах, что может быть полезно при их программной и аппаратной реализации для высокопроизводительных систем.

Авторы выражают искреннюю благодарность анонимному рецензенту за исключительно глубокий и содержательный анализ. Столь внимательное рецензирование существенно повысило научную ценность и строгость нашей работы.

ЛИТЕРАТУРА

1. *Ullrich M., De Cannière, C., Indesteege S., et al.* Finding Optimal Bitsliced Implementations of 4 x 4-bit S-boxes. Ecrypt II. 2011. http://skew2011.mat.dtu.dk/proceedings/Finding Optimal Bitsliced Implementations of 4 to 04-bit S-boxes.pdf.
2. *Чичаева А. А.* Поиск эффективно реализуемых подстановок с оптимальными криптографическими характеристиками. Рускрипто'21. Солнечногорск, 2021. https://ruscrypto.ru/resource/archive/rc2021/files/02_chichayeva.pdf.
3. *Jean J., Peyrin T., Sim S. M., and Tourteaux J.* Optimizing implementations of lightweight building blocks // IACR Trans. Symmetric Cryptol. 2017. V. 4. P. 130–168.
4. *Rudell R. L.* Multiple-Valued Logic Minimization for PLA Synthesis. Technical Report. EECS Department, University of California, 1986. https://www2.eecs.berkeley.edu/Pubs/TechRpts/1986/ERL-86-65.pdf.
5. *Hlavička J. and Fičer P.* A heuristic Boolean minimizer // Proc ICCAD'01. San Jose, California, 2001. P. 439–442.
6. *Avraamova O. D., Fomin D. B., Serov V. A., et al.* A compact bit-sliced representation of Kuznyechik S-box // Матем. вопр. криптогр. 2021. Т. 12. № 2. С. 21–38.
7. *Dansarie M.* sboxgates: A program for finding low gate count implementations of s-boxes // J. Open Source Software. 2021. No. 6(62). https://joss.theoj.org/papers/10.21105/joss.02946.
8. *Nogami Y., Nekado K., Toyota T., et al.* Mixed bases for efficient inversion in $\mathrm{GF}((2^2)^2)^2$ and conversion matrices of subbytes of AES // LNCS. 2010. V. 6225. P. 234–247.
9. *Turan M. S., McKay K., Chang D., et al.* Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. NIST, 2021. https://doi.org/10.6028/NIST.IR.8369.
10. *Сэвидж Д. Э.* Сложность вычислений. М.: Факториал, 1998. 368 с.
11. *Mano M. M. and Kime C.* Logic and Computer Design Fundamentals. Prentice Hall Press, 2007. 672 p.
12. *Boot T. L.* Digital Networks and Computer Systems. N.Y.: Wiley, 1971. 451 p.
13. *Лупанов О. Б.* О реализации функции алгебры логики формулами из конечных классов (формулами ограниченной глубины) в базисе $\&, \vee, {}^-$ // Проблемы кибернетики. 1961. Т. 6. С. 5–14.
14. *Canteaut A. and Perrin L.* On CCZ-equivalence, extended-affine equivalence, and function twisting // Finite Fields Appl. 2018. V. 56. P. 209–246.
15. *Olofsson M.* VLSI Aspects on Inversion in Finite Fields. PhD Thesis. Linköping, Sweden, 2002.
16. *Drolet G.* A new representation of elements of finite fields $\mathrm{GF}(2^m)$ yielding small complexity arithmetic circuits // IEEE Trans. Computers. 1998. V. 47. No. 9. P. 938–946.
17. *Wu H.* Low complexity bit-parallel finite field arithmetic using polynomial basis // LNCS. 1999. V. 1717. P. 280–291.
18. *Ueno R., Homma N., Nogami Y., et al.* Highly efficient $\mathrm{GF}(2^8)$ inversion circuit based on hybrid GF representations // J. Cryptogr. Engin. 2019. V. 9. No. 2. P. 101–113.
19. *Nekado K., Nogami Y., and Iokibe K.* Very short critical path implementation of AES with direct logic gates // LNCS. 2012. V. 7631. P. 51–68.
20. *Fomin D. B.* New classes of 8-bit permutations based on a butterfly structure // Матем. вопр. криптогр. 2019. Т. 10. № 2. С. 169–180.
21. *Фомин Д. Б.* Построение подстановок пространства V_{2m} с использованием $(2m, m)$-функций // Матем. вопр. криптогр. 2020. Т. 11. № 3. С. 121–138.

22. *Фомин Д. Б.* Об алгебраической степени и дифференциальной равномерности подстановок пространства V_{2m}, построенных с использованием $(2m, m)$-функций // Матем. вопр. криптогр. 2020. Т. 11. № 4. С. 133–149.
23. *Fomin D. B. and Kovrizhnykh M. A.* On differential uniformity of permutations derived using a generalized construction // Матем. вопр. криптогр. 2022. Т. 13. № 2. С. 37–52.
24. *Фомин Д. Б., Трифонов Д. И.* Об аппаратной реализации одного класса байтовых подстановок // Прикладная дискретная математика. Приложение. 2019. № 12. С. 134–137.
25. *Rebeiro C., Selvakumar A. D., and Devi A. S. L.* Bitslice implementation of AES // LNCS. 2006. V. 4301. P. 203–212.
26. *Grosso V., Leurent G., Standaert F., and Varici K.* LS-designs: Bitslice encryption for efficient masked software implementations // LNCS. 2014. V. 8540. P. 18–37.
27. *Ковржных М. А., Фомин Д. Б.* Об эвристическом алгоритме построения подстановок с заданными криптографическими характеристиками с использованием обобщённой конструкции // Прикладная дискретная математика. 2022. Т. 57. С. 5–21.
28. *Saarinen M-J. O.* Cryptographic analysis of all 4 x 4-bit S-Boxes // LNCS. 2011. V. 7118. P. 118–133.
29. *Biryukov A., Perrin L., and Udovenko A.* Reverse-engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1 // LNCS. 2016. V. 9665. P. 372–402.
30. *De la Cruz Jiménez R. A.* Generation of 8-bit s-boxes having almost optimal cryptographic properties using smaller 4-bit s-boxes and finite field multiplication // LNCS. 2017. V. 11368. P. 191–206.
31. *Lidl R. and Niederreiter H.* Finite Fields. Cambridge: Cambridge University Press, 1997. 755 p.
32. *Canright D.* A very compact s-box for AES // LNCS. 2005. V. 3659. P. 441–455.
33. *Morioka S. and Satoh A* An optimized s-box circuit architecture for low power AES design // LNCS. 2002. V. 2523. P. 172–186.
34. *Itoh T. and Tsujii S.* A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases // Information and Computation. 1988. V. 78. No. 3. P. 171–177.
35. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the Bit-Slice Representations of Some Nonlinear Bijective Transformations. CTCrypt'23. Волгоград, 2023. `https://ctcrypt.ru/files/files/2023/08/03.pdf`.
36. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the Bit-Slice representations of some nonlinear bijective transformations // Матем. вопр. криптогр. 2024. Т. 15. № 1. С. 97–125.

REFERENCES

1. *Ullrich M., De Cannière, C., Indesteege S., et al.* Finding Optimal Bitsliced Implementations of 4 x 4-bit S-boxes. Ecrypt II. 2011. `http://skew2011.mat.dtu.dk/proceedings/Finding Optimal Bitsliced Implementations of 4 to 04-bit S-boxes.pdf`.
2. *Chichaeva A. A.* Poisk effektivno realizuemykh podstanovok s optimal'nymi kriptograficheskimi kharakteristikami [Search for Efficiently Implementable Substitutions with Optimal Cryptographic Characteristics]. Ruskripto'21, Solnechnogorsk, 2021. `https://ruscrypto.ru/resource/archive/rc2021/files/02_chichayeva.pdf`. (in Russian)
3. *Jean J., Peyrin T., Sim S. M., and Tourteaux J.* Optimizing implementations of lightweight building blocks. IACR Trans. Symmetric Cryptol., 2017, vol. 4, pp. 130–168.
4. *Rudell R. L.* Multiple-Valued Logic Minimization for PLA Synthesis. Technical Report, EECS Department, University of California, 1986. `https://www2.eecs.berkeley.edu/Pubs/TechRpts/1986/ERL-86-65.pdf`.

5. *Hlavička J. and Fičer P.* A heuristic Boolean minimizer. Proc ICCAD'01, San Jose, California, 2001, pp. 439–442.
6. *Avraamova O. D., Fomin D. B., Serov V. A., et al.* A compact bit-sliced representation of Kuznyechik S-box. Matematicheskie Voprosy Kriptografii, 2021, vol. 12, no. 2, pp. 21–38.
7. *Dansarie M.* sboxgates: A program for finding low gate count implementations of s-boxes. J. Open Source Software, 2021, no. 6(62). `https://joss.theoj.org/papers/10.21105/joss.02946`.
8. *Nogami Y., Nekado K., Toyota T., et al.* Mixed bases for efficient inversion in $\mathrm{GF}((2^2)^2)^2$ and conversion matrices of subbytes of AES. LNCS, 2010, vol. 6225, pp. 234–247.
9. *Turan M. S., McKay K., Chang D., et al.* Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. NIST, 2021. `https://doi.org/10.6028/NIST.IR.8369`.
10. *Savage J. E.* The Complexity of Computing. N.Y., Wiley, 1976. 424 p.
11. *Mano M. M. and Kime C.* Logic and Computer Design Fundamentals. Prentice Hall Press, 2007. 672 p.
12. *Boot T. L.* Digital Networks and Computer Systems. N.Y., Wiley, 1971. 451 p.
13. *Lupanov O. B.* O realizatsii funktsii algebry logiki formulami iz konechnykh klassov (formulami ogranichennoy glubiny) v bazise $\&, \vee, {}^-$ [On implementation of a function of logic algebra by formulas from finite classes (formulas of bounded depth) in a basis $\&, \vee, {}^-$]. Problemy Kibernetiki, 1961, vol. 6, pp. 5–14. (in Russian)
14. *Canteaut A. and Perrin L.* On CCZ-equivalence, extended-affine equivalence, and function twisting. Finite Fields Appl., 2018, vol. 56, pp. 209–246.
15. *Olofsson M.* VLSI Aspects on Inversion in Finite Fields. PhD Thesis, Linköping, Sweden, 2002.
16. *Drolet G.* A new representation of elements of finite fields $\mathrm{GF}(2^m)$ yielding small complexity arithmetic circuits. IEEE Trans. Computers, 1998, vol. 47, no. 9, pp. 938–946.
17. *Wu H.* Low complexity bit-parallel finite field arithmetic using polynomial basis. LNCS, 1999, vol. 1717, pp. 280–291.
18. *Ueno R., Homma N., Nogami Y., et al.* Highly efficient $\mathrm{GF}(2^8)$ inversion circuit based on hybrid GF representations. J. Cryptogr. Engin., 2019, vol. 9, no. 2, pp. 101–113.
19. *Nekado K., Nogami Y., and Iokibe K.* Very short critical path implementation of AES with direct logic gates. LNCS, 2012, vol. 7631, pp. 51–68.
20. *Fomin D. B.* New classes of 8-bit permutations based on a butterfly structure. Matematicheskie Voprosy Kriptografii, 2019, vol. 10, no. 2, pp. 169–180.
21. *Fomin D. B.* Postroenie podstanovok prostranstva V_{2m} s ispol'zovaniem $(2m, m)$-funktsiy [Construction of permutations on the space V_{2m} by means of $(2m, m)$-functions]. Matematicheskie Voprosy Kriptografii, 2020, vol. 11, no. 3, pp. 121–138. (in Russian)
22. *Fomin D. B.* Ob algebraicheskoy stepeni i differentsial'noy ravnomernosti podstanovok prostranstva V_{2m}, postroennykh s ispol'zovaniem $(2m, m)$-funktsiy [On the algebraic degree and differential uniformity of permutations on the space V_{2m} constructed via $(2m, m)$-functions]. Matematicheskie Voprosy Kriptografii, 2020, vol. 11, no. 4, pp. 133–149. (in Russian)
23. *Fomin D. B. and Kovrizhnyh M. A.* On differential uniformity of permutations derived using a generalized construction. Matematicheskie Voprosy Kriptografii, 2022, vol. 13, no. 2, pp. 37–52.
24. *Fomin D. B. and Trifonov D. I.* Ob apparatnoy realizatsii odnogo klassa baytovykh podstanovok [Hardware implementation of one class of 8-bit permutations]. Prikladnaya diskretnaya matematika. Prilozhenie, 2019. vol. 12, pp. 134–137. (in Russian)

25. *Rebeiro C., Selvakumar A. D., and Devi A. S. L.* Bitslice implementation of AES. LNCS, 2006, vol. 4301, pp. 203–212.
26. *Grosso V., Leurent G., Standaert F., and Varici K.* LS-designs: Bitslice encryption for efficient masked software implementations. LNCS, 2014, vol. 8540, pp. 18–37.
27. *Kovrizhnykh M. A. and Fomin D. B.* Ob evristicheskom algoritme postroeniya podstanovok s zadannymi kriptograficheskimi kharakteristikami s ispol'zovaniem obobshchennoy konstruktsii [Heuristic algorithm for obtaining permutations with given cryptographic properties using a generalized construction]. Prikladnaya Diskretnaya Matematika, 2022, vol. 57, pp. 5–21. (in Russian)
28. *Saarinen M-J. O.* Cryptographic analysis of all 4 x 4-bit S-Boxes. LNCS, 2011, vol. 7118, pp. 118–133.
29. *Biryukov A., Perrin L., and Udovenko A.* Reverse-engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1. LNCS, 2016, vol. 9665, pp. 372–402.
30. *De la Cruz Jiménez R. A.* Generation of 8-bit s-boxes having almost optimal cryptographic properties using smaller 4-bit s-boxes and finite field multiplication. LNCS, 2017, vol. 11368, pp. 191–206.
31. *Lidl R. and Niederreiter H.* Finite Fields. Cambridge, Cambridge University Press, 1997. 755 p.
32. *Canright D.* A very compact s-box for AES. LNCS, 2005, vol. 3659, pp. 441–455.
33. *Morioka S. and Satoh A* An optimized s-box circuit architecture for low power AES design. LNCS, 2002, vol. 2523, pp. 172–186.
34. *Itoh T. and Tsujii S.* A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. Information and Computation, 1988, vol. 78, no. 3, pp. 171–177.
35. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the Bit-Slice Representations of Some Nonlinear Bijective Transformations. CTCrypt'23, Volgograd, 2023. `https://ctcrypt.ru/files/files/2023/08/03.pdf`.
36. *Puente O. C., Leal R. F., and de la Cruz Jiménez R. A.* On the bit-slice representations of some nonlinear bijective transformations. Matematicheskie Voprosy Kriptografii, 2024, vol. 15, no. 1, pp. 97–125.

Приложение. Реализация подстановки π_R

```
def bitAlpha(inp):
    out2 = inp[4] ^ inp[7]
    out3 = inp[6] ^ inp[2]
    out0 = out3 ^ inp[5]
    out7 = inp[6] ^ inp[0]
    tmp1   = inp[3] ^ out7
    out6 = tmp1 ^ inp[1]
    out1 = inp[7] ^ inp[5] ^ inp[3] ^ inp[1]
    out4 = out1 ^ inp[2]
    out5 = tmp1 ^ out4
    return [out0, out1, out2, out3, out4, out5, out6, out7]

def bitOmega(inp):
    out1 = inp[1] ^ inp[7]
    out2 = inp[2] ^ inp[6]
    out7 = inp[3] ^ inp[1]
    out4 = inp[4] ^ inp[7] ^ out7
    out0 = inp[0]
    out3 = inp[3]
    out5 = inp[5]
    out6 = inp[2]
```

```
    return [out0, out1, out2, out3, out4, out5, out6, out7]

def bitInv(inp):
    tmp1   = inp[1] & inp[2]
    tmp2   = (tmp1 ^ inp[0]) & inp[3]
    out2 = inp[2] ^ tmp2
    tmp3   = tmp1 | inp[3]
    out3 = tmp3 ^ tmp2 ^ (tmp2 | inp[2])
    out1 = ((inp[1] & inp[3]) ^ out3) ^ (tmp1 | inp[0])
    out0 = inp[1] ^ out1 ^ tmp3 ^ (out1 & inp[0])
    return [out0, out1, out2, out3]

def bitNu0(inp):
    tmp1   = inp[1] & inp[2]
    tmp2   = tmp1 ^ inp[0]
    tmp3   = inp[1] ^ tmp2
    out1 = tmp2 ^ ((tmp3 | inp[2]) & inp[3])
    tmp4   = (tmp1 | tmp2) ^ 1
    tmp5   = inp[1] & tmp3
    out0 = tmp4 ^ ((tmp5 ^ inp[2]) | inp[3])
    tmp6   = out0 & out1
    out3 = tmp5 ^ ((tmp6 ^ inp[2]) & inp[3])
    out2 = ((tmp6 | tmp4) ^ tmp3) ^ inp[3]
    return [out0, out1, out2, out3]

def bitNu1(inp):
    out3 = (inp[3] | inp[1]) ^ inp[2] ^ inp[0]
    tmp1   = inp[1] ^ 1
    out0 = tmp1 ^ (out3 | inp[3])
    tmp2   = out3 ^ inp[1]
    out1 = (tmp2 | out0) ^ inp[2]
    out2 = (tmp2 | tmp1) ^ inp[3]
    return [out0, out1, out2, out3]

def bitPhi(inp):
    tmp1   = inp[1] ^ inp[3]
    tmp2   = inp[1] | inp[0]
    tmp3   = (tmp2 & tmp1) | inp[2]
    out2 = tmp1 ^ inp[0] ^ tmp3
    tmp4   = inp[2] ^ tmp2
    tmp5   = (tmp4 & out2) | inp[3]
    out3 = inp[0] ^ tmp5
    tmp6   = (inp[1] ^ tmp5) & tmp3
    out1 = tmp6 ^ (tmp4 & inp[1])
    out0 = (tmp4 ^ 1) | tmp6
    return [out0, out1, out2, out3]

def bitSigma(inp):
    tmp1   = inp[3] ^ (inp[2] | inp[1])
    out1 = tmp1 ^ (((inp[2] & tmp1) ^ inp[1]) | inp[0])
    tmp2   = inp[0] ^ tmp1
    tmp3   = tmp2 ^ inp[1]
    tmp4   = tmp1 ^ tmp3
    tmp5   = tmp4 & tmp2
    tmp6   = tmp5 | inp[2]
    out0 = tmp3 ^ tmp6
    out2 = tmp4 ^ ((tmp5 ^ 1 ^ inp[2]) | inp[0])
    out3 = (tmp2 | out2) ^ tmp6 ^ inp[3]
    return [out0, out1, out2, out3]
```

```
def mulf_2_2(a,b):
    tmp1 = a[0] ^ a[1]
    tmp2 = b[0] ^ b[1]
    tmp3 = a[0] & b[0]
    tmp4 = tmp1 & tmp2
    tmp5 = b[1] & a[1]
    out0 = tmp3 ^ tmp4
    out1 = tmp5 ^ tmp4
    return [out0, out1]

def bitAlpha4(a):
    out0 = a[0] ^ a[1]
    out1 = a[0]
    return [out0, out1]

def mul2_2_2(a, b):
    tmp10 = a[0] ^ a[2]
    tmp11 = a[1] ^ a[3]
    tmp20 = b[0] ^ b[2]
    tmp21 = b[1] ^ b[3]
    tmp3    = mulf_2_2([a[0], a[1]], [b[0], b[1]])
    tmp4    = mulf_2_2([tmp10, tmp11], [tmp20,tmp21])
    tmp5    = mulf_2_2([a[2], a[3]], [b[2], b[3]])
    tmp6    = bitAlpha4(tmp5)
    out0 = tmp3[0] ^ tmp6[0]
    out1 = tmp3[1] ^ tmp6[1]
    out2 = tmp3[0] ^ tmp4[0]
    out3 = tmp3[1] ^ tmp4[1]
    return [out0, out1, out2, out3]

def bitInd(inp):
    return (inp[0] | inp[1] | inp[2] | inp[3]) ^ 1

def pi_r(x):
    tmp1 = bitAlpha(x)
    r = [tmp1[0], tmp1[1], tmp1[2], tmp1[3]]
    l = [tmp1[4], tmp1[5], tmp1[6], tmp1[7]]
    tmp2 = bitInd(r)
    tmp3 = bitNu0(l)
    tmp4 = bitNu1(mul2_2_2(l, bitInv(r)))
    l[0] = (tmp2 & tmp3[0]) ^ tmp4[0]
    l[1] = (tmp2 & tmp3[1]) ^ tmp4[1]
    l[2] = (tmp2 & tmp3[2]) ^ tmp4[2]
    l[3] = (tmp2 & tmp3[3]) ^ tmp4[3]
    r = bitSigma(mul2_2_2(r,bitPhi(l)))
    out = bitOmega(r+l)
    return out

pi = []
for x in range(256):
    inp = [int(t) for t in bin(x)[2:].zfill(8)][::-1]
    tmp = pi_r(inp)
    res = 0
    for i in range(len(tmp)):
        res = res + (tmp[i] << i)
    pi.append(res)
print(pi)
```