НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
май–июнь 2026 Том 26 № 3 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
May–June 2026 Vol. 26 No 3 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2026-26-3-617-627
УДК 004.056

Алгоритм генерации криптографически стойких S-блоков, оптимизированных для программируемых логических интегральных схем

Илья Владиславович Шибакин[1], Алла Борисовна Левина[2]✉, Михаил Степанович Куприянов[3]

[1,2,3] Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197376, Российская Федерация

[1] shibakin12@mail.ru, https://orcid.org/0009-0002-8692-7474
[2] Alla_levina@mail.ru✉, https://orcid.org/0000-0003-4421-2411
[3] mskupriyanov@etu.ru, https://orcid.org/0000-0003-4695-4507

Аннотация

Введение. Криптографические примитивы, реализованные на программируемых логических интегральных схемах (ПЛИС), обладают рядом преимуществ по сравнению со своими программными аналогами, они эффективнее по производительности и потреблению энергии. В то же время, для реализации на ПЛИС, важна компактность представления элементов криптографического примитива, например таблиц замен (S-блоков). В работе представлен алгоритм генерации криптографически стойких S-блоков, компактных в реализации на ПЛИС. **Метод.** На основе алгоритма генерации бент-функций с помощью вейвлет-преобразования разработан алгоритм построения S-блоков. Рассчитаны нелинейность, дифференциальная равномерность, лавинный эффект, критерий независимости бит, генерируемых предложенным алгоритмом S-блоков. Проведено сравнение криптографических характеристик генерируемых S-блоков и S-блоков, используемых в американском стандарте симметричного шифрования Advanced Encryption Standard и отечественном шифре «Кузнечик». Проанализирован лавинный эффект хэш-функции «Стрибог» при использовании S-блока и S-блока, сгенерированного разработанным алгоритмом, проведено сравнение затрат ресурсов ПЛИС. **Основные результаты.** По криптографическим характеристикам генерируемые разработанным алгоритмом S-блоки превосходят, или не уступают существующим, но не являются биективными, что сужает сферу их применения. Лавинный эффект хэш-функции «Стрибог» при применении ее S-блока и S-блока, сгенерированного разработанным алгоритмом, не отличаются. Анализ результатов синтеза реализаций хэш-функции показал, что использование разработанного S-блока позволяет экономить около 1000 Look Up Table на ПЛИС. **Обсуждение.** Разработанные S-блоки по криптографическим характеристикам не уступают существующим, или превосходят их. Предложенные S-блоки компактнее использующихся в аппаратной реализации на ПЛИС, но не биективны, что делает наиболее целесообразным и естественным их применение в необратимых криптографических примитивах, реализованных на ресурсо-ограниченных встраиваемых системах.

Ключевые слова

S-блок, ПЛИС, бент-функция, лавинный эффект, нелинейность, дифференциальная равномерность, BIC

Благодарности

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации № 075-00003-24-02 от 08.02.2024 (проект FSEE-2024-0003).

Ссылка для цитирования: Шибакин И.В., Левина А.Б., Куприянов М.С. Алгоритм генерации криптографически стойких S-блоков, оптимизированных для программируемых логических интегральных схем // Научно-технический вестник информационных технологий, механики и оптики. 2026. Т. 26, № 3. С. 617–627. doi: 10.17586/2226-1494-2026-26-3-617-627

An algorithm for generating cryptographically stable S-box optimized for programmable logic integrated circuits

Ilya V. Shibakin[1], Alla B. Levina[2]✉, Mikhail S. Kupriyanov[3]

[1,2,3] Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, 197376, Russian Federation

[1] shibakin12@mail.ru, https://orcid.org/0009-0002-8692-7474
[2] Alla_levina@mail.ru✉, https://orcid.org/0000-0003-4421-2411
[3] mskupriyanov@etu.ru, https://orcid.org/0000-0003-4695-4507

Abstract

Cryptographic primitives implemented on Field-Programmable Gate Array (FPGA) have a number of advantages over their software counterparts; they are more efficient in terms of performance and energy consumption. At the same time, for implementation on FPGAs, it is important to have a compact representation of cryptographic primitive elements such as substitution tables (S-boxes). This paper presents a cryptographically strong S-box that is compact in its implementation on FPGAs. Based on the algorithm for generating bent functions using wavelet transform, an algorithm for constructing S-boxes has been developed. The nonlinearity, differential uniformity, avalanche effect, and Bit Independence Criterion (BIC) of the S-boxes generated by the developed algorithm are calculated. Diagrams have been constructed showing the distribution of S-boxes by nonlinearity and differential uniformity values for different values of the algorithm input data. A comparison was made of the cryptographic properties of the generated S-boxes and the S-boxes used in the American Advanced Encryption Standard symmetric encryption standard and the Russian Kuznechik cipher. The avalanche effect of the Streebog hash function was analyzed when using its S-box and the S-box generated by the developed algorithm, and a comparison of FPGA resource utilization was performed. The best results in terms of nonlinearity, differential uniformity, avalanche effect, and BIC are achieved by S-boxes consisting of bent functions that have no terms higher than degree two (quadratic bent functions) in algebraic normal form. In terms of cryptographic properties, the S-boxes generated by the developed algorithm are superior to or not inferior to existing ones, but they are not bijective, which narrows their field of application. The avalanche effect of the hash function with its original S-box and the S-box generated by the developed algorithm do not differ. Analysis of the results of the synthesis of hash function implementations showed that the use of the developed S-box allows saving about 1000 Look Up Tables on the FPGA. The S-boxes presented in this work are not inferior to existing ones in terms of cryptographic properties, or surpass them. The developed S-boxes are more compact than existing ones in hardware implementation on FPGAs, but they are not bijective, which makes their use most appropriate and natural in irreversible cryptographic primitives implemented on resource-limited embedded systems.

Keywords

S-box, FPGA, bent function, avalanche effect, nonlinearity, differential uniformity, BIC

Acknowledgements

This research was funded by the Ministry of Science and Higher Education of the Russian Science Foundation (Project "Goszaadanie" No. 075-00003-24-02, FSEE-2024-0003).

For citation: Shibakin I.V., Levina A.B., Kupriyanov M.S. An algorithm for generating cryptographically stable S-box optimized for programmable logic integrated circuits. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2026, vol. 26, no. 3, pp. 617–627 (in Russian). doi: 10.17586/2226-1494-2026-26-3-617-627

Введение

S-блоки являются основной частью многих современных криптографических симметричных шифров, примерами могут служить Data Encryption Standard (DES), Advanced Encryption Standard (AES), «Магма», «Кузнечик». Отметим, что данные шифры, также используются в хэш-функциях, например, «Стрибог».

С точки зрения математики S-блок представляет векторную булеву функцию, имеющую n входных и m выходных бит. Векторные булевы функции с подходящими свойствами выполняют в криптографии роль нелинейного преобразования.

Качество S-блоков, используемых в криптографических примитивах, важно для криптографической стойкости в целом. Например, S-блоки шифра DES были уязвимы для линейного криптоанализа в силу низкой нелинейности функций, использованных в них. В работе [1] предложен метод линейного криптоанализа, применяющий слабые стороны конструкции S-блоков DES. В [2] показан один из примеров метода построения S-блоков, где предложен простой способ формирования S-блоков с использованием булевых функций и перестановок. В работе [2] рассмотрены вектора булевых функций, к аргументам которых различное число раз применяется заданная перестановка. Варьируя перестановки и параметры применяемой скалярной булевой функции, выполнено исследование свойств предложенной конструкции, такие как сбалансированность, нелинейность, дифференциальную равномерность. Результаты позволяют предполагать, что предложенная конструкция даст возможность создавать векторные булевы функции произвольного числа переменных с хорошими криптографическими свойствами. Однако в [2] рассмотрены перестановки и вектора функций только небольшой размерности (от 2 до 4), т. е. свойства векторов произвольного размера от произвольного числа переменных не рассматривались.

В работе [3] применен алгебраический подход к построению S-блока и предложена биективная функция над конечным полем, выражающая S-блок. Также в [3] выполнена оценка и сравнение с S-блоком AES и S-блоком, предложенным в исследовании [4]. Сравнение проводилось по ряду криптографических

характеристик, таких как нелинейность, дифференциальная равномерность, максимальный период элемента в S-блоке, алгебраическая сложность, лавинный эффект, корреляция между выходными битами блока. По всем параметрам S-блок из [3] оказался по сравнению с существующими.

Достаточно сильными криптографическими свойствами обладает S-блок, построенный на кубическом полиномиальном отображении [5].

В отличие от исследований [2–5], в настоящей работе для генерации S-блоков используется подход, дающий возможность задания параметров генерируемых блоков и степени нелинейности булевых функций, из которых они состоят. Главным преимуществом генерируемых S-блоков является их ориентированность на реализацию с помощью программируемых логических интегральных схем (ПЛИС).

Одной из ключевых особенностей разработанного алгоритма генерации S-блоков является применение в его составе бент-функций.

Американский математик Оскар Ротхаус ввел понятие бент-функций и начал их исследование в 60-х годах прошлого века [6]. В 70-х годах над изучением этой области работали Дж. Диллон [7] и Р.Л. МакФарланд [8], а к 80-м интерес к бент-функциям возрос, и их изучением занялись ученые-математики по всему миру.

Значительный вклад в современное состояние теории бент-функций внес Клод Карле, чьи фундаментальные работы, в том числе [9] позволили систематизировать их свойства, разработать новые конструкции и исследовать связи с теорией кодирования и криптографией. В частности, исследования Карле гипер-бент-функций и функций вида $tr(ax^d)$ заложили теоретический фундамент для разработки эффективных алгоритмов генерации, подобных примененному в данной работе.

В настоящей работе представлен алгоритм генерации S-блоков, использующий вейвлет-преобразование. Генерируемые алгоритмом S-блоки обладают высокой криптографической стойкостью и ориентированы на применение на ПЛИС. Приведен пример реализации на ПЛИС хэш-функции «Стрибог» с применением разработанного S-блока, доказывающий эффективность предложенного подхода.

Методы

S-блок и его характеристики. Напомним, что S-блок — векторная булева функция f: $\mathbb{Z}_2^n \to \mathbb{Z}_2^m$, также называемая (n, m)-функция. Таким образом, m определяет количество функций, составляющих S-блок, а n — количество переменных в них.

Рассмотрим наиболее важные характеристики S-блоков.

S-блок, в составе примитива, выполняет функцию нелинейного преобразования, т. е. одним из важнейших свойств S-блока является невозможность представить его линейными функциями. Для произвольной не векторной функции f нелинейность N_f, выраженная через коэффициенты Уолша–Адамара $W_f(\mathbf{v}) = \sum\limits_{\mathbf{u}\in\mathbb{Z}_2^n} (-1)^{(\mathbf{u}\cdot\mathbf{v})\oplus f(\mathbf{u})}$, рассчитывается так:

$$N_f = 2^{n-1} - \frac{1}{2}\max_{\mathbf{v}\in\mathbb{Z}_2^n} |W_f(\mathbf{v})|,$$

где $\mathbf{v}$ и $\mathbf{u}$ — произвольные битовые вектора длины n.

Нелинейность векторной булевой функции ограничена сверху значением $2^{n-1} - 2^{n/2-1}$. Векторная булева функция является векторной бент-функцией, если ее нелинейность имеет максимальное значение.

Для (n, m)-функции f преобразование Уолша–Адамара W_f^{vest}: $\mathbb{Z}_2^n \times \mathbb{Z}_2^m \to \mathbb{Z}$ имеет следующий вид:

$$W_f^{vest}(\mathbf{a}, \mathbf{b}) = \sum_{\mathbf{v}\in\mathbb{Z}_2^n} (-1)^{(\mathbf{a}\cdot\mathbf{v})\oplus(\mathbf{b}f(\mathbf{v}))},$$

где $\mathbf{a}$ — произвольный битовый вектор длины n; $\mathbf{b}$ — произвольный битовый вектор длины m.

Нелинейность (n, m)-функции f является минимальной из нелинейностей $f_{\mathbf{b}}$, где $f_{\mathbf{b}}(\mathbf{v}) = \mathbf{b}f(\mathbf{v})$ представляет из себя линейную комбинацию функций, составляющих f:

$$N_f = \min_{\mathbf{b}\in(\mathbb{Z}_2^m)\setminus\{0\}} dist(f_{\mathbf{b}}, A_n) =$$
$$= 2^{n-1} - \frac{1}{2}\max_{\mathbf{a}\in\mathbb{Z}_2^n,\mathbf{b}\in(\mathbb{Z}_2^m)\setminus\{0\}} |W_f^{vest}(\mathbf{a}, \mathbf{b})|,$$

где A_n — множество аффинных векторных функций.

Стойкость S-блока к дифференциальному криптоанализу [10] определяет дифференциальная равномерность, вычисляемая как

$$\max_{\mathbf{a}\in\mathbb{Z}_2^n,\mathbf{b}\in(\mathbb{Z}_2^n)^*} \delta_{f_{\mathbf{a},\mathbf{b}}},$$

где $\delta_{f_{\mathbf{a},\mathbf{b}}} = |\{\mathbf{v} \in \mathbb{Z}_2^n | f(\mathbf{v}) \oplus f(\mathbf{v} \oplus \mathbf{a}) = \mathbf{b}\}|$.

Наименьшее возможное значение $\delta_{f_{\mathbf{a},\mathbf{b}}}$ равно 2 и если оно достигается, то (n,n)-функция принадлежит к классу почти совершенно нелинейных (Almost Perfectly Nonlinear) [11].

Для S-блока существует понятие лавинного эффекта (Avalanche Effect, AE), методы его оценки приведены в работах [3, 12].

Лавинный эффект рассчитывается следующим образом:

$$AE_i = \sum_{\mathbf{v}\in\mathbb{Z}_2^n} hwt(f(\mathbf{v}) \oplus f(\mathbf{v} \oplus \mathbf{C}_{\mathbf{n}}^{\mathbf{i}})),$$

где hwt — вес Хэмминга; $\mathbf{C}_{\mathbf{n}}^{\mathbf{i}}$ — битовый вектор длины n, где все позиции, кроме i-й заняты нулями, i от 1 до n.

В лучшем случае $\forall i$, $1 \le i \le n$, $AE_i = m2^{n-1}$, т. е. в среднем при изменении одного входного бита меняется половина бит на выходе.

Более требовательным к качеству S-блока показателем является «строгий» критерий лавинного эффекта (Strict Avalanche Criterion, SAC), впервые предложенный в [13]. Элементы матрицы SAC, выражают для S-блока вероятность изменения выходного бита с номером i при изменении входного бита j:

$$SAC_{i,j} = \frac{1}{2^n}\sum_{\mathbf{v}\in\mathbb{Z}_2^n} f_i(\mathbf{v}) \oplus f_i(\mathbf{v} \oplus \mathbf{C}_{\mathbf{n}}^{\mathbf{j}}).$$

Отметим, что в формуле вычисления $SAC_{i,j}$ знак суммы обозначает алгебраическую сумму, а не «ис-

ключающее ИЛИ» (XOR) значений. Каждое слагаемое в формуле вычисления $SAC_{i,j}$ представляет двоичное значение (1 или 0), так как является результатом XOR значений обычных (не векторных) булевых функций.

Для вычисления $SAC_{i,j}$ необходимо знание таблиц истинности булевых функций f_i, составляющих S-блок. Таблицы истинности можно рассчитать на основе самого S-блока f. Каждое m-битное значение $f(\mathbf{v})$ представляется в виде вектора $(f_1(\mathbf{v}) \ldots f_m(\mathbf{v}))$.

В случае выполнения SAC все значения в матрице SAC будут равны 0,5, при изменении любого входного бита каждый из выходных бит будет меняться с вероятностью 0,5.

Существует и более компактная форма выражения степени соответствия S-блока критерию лавинного эффекта — Distance to SAC (DSAC):

$$DSAC(f) = \sum_{i\in\overline{1,m}} \sum_{j\in\overline{1,n}} \left| \sum_{\mathbf{v}\in\mathbb{Z}_2^n} f_i(\mathbf{v}) \oplus f_i(\mathbf{v} \oplus \mathbf{C}_{\mathbf{n}}^{\mathbf{j}}) - 2^{n-1} \right|.$$

Важной характеристикой для S-блока также является критерий независимости битов (Bit Independence Criterion, BIC) [14]. Его численное выражение характеризует степень зависимости между выходными битами:

Обозначим

$$f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{C}_{\mathbf{n}}^{\mathbf{i}}) = (a_{i,0}(\mathbf{x}), \ldots, a_{i,n-1}(\mathbf{x}))$$

для любого произвольного битового вектора $\mathbf{x}$ длины n, тогда элементы матрицы BIC выражаются так

$$BIC_{j,k} = \max_{0\le i\le n-1} corr(a_{i,j}, a_{i,k}), j \in \overline{0, n-1}, k \in \overline{0, n-1},$$

где $a_{i,j}$ — последовательность $a_{i,j}(x)$ для всех возможных значений $\mathbf{x}$, индексы j и k обозначают номера выходных бит, а корреляция рассчитывается по формуле:

$$corr(a_{i,j}, a_{i,k}) = \frac{cov(a_{i,j}, a_{i,k})}{\sqrt{D}(a_{i,j})\sqrt{D}(a_{i,k})} =$$

$$= \frac{E(a_{i,j}\cdot a_{i,k}) - E(a_{i,j})E(a_{i,k})}{\sqrt{E(a_{i,j}^2) - E^2(a_{i,j})}\sqrt{E(a_{i,k}^2) - E^2(a_{i,k})}},$$

где $E(t)$ — математическое ожидание величины t; $D(t)$ — ее дисперсия.

Предложенный алгоритм генерации S-блоков

Разработанный алгоритм, генерирующий S-блоки, основан на алгоритме генерации бент-функций с помощью вейвлет-преобразования, описанном в работе [15].

Бент-функцией называется булева функция f с четным числом переменных n, имеющая максимальное значение нелинейности N_f [11], т. е.

$$N_f = 2^{n-1} - 2^{n/2-1}.$$

Вейвлет-преобразование применяется в самом общем случае для обработки числовых потоков и призвано сократить объем передаваемой информации, путем отбрасывания той ее части, которая с какой-либо точки зрения является несущественной.

Идея вейвлет-преобразования заключается в разделении информационного потока c_j на основной a_j и вейвлетный b_j, служащий для уточнения [16]:

$$a_j = \frac{c_{2j} + c_{2j+1}}{2}, \quad b_j = \frac{c_{2j} - c_{2j+1}}{2}.$$

Для восстановления информационного потока используются следующие формулы:

$$c_{2j} = a_j + b_j, \quad c_{2j+1} = a_j - b_j.$$

Исходный поток разделяется на основной и вейвлетный согласно формулам декомпозиции, а переход от двух потоков к исходному происходит по формулам реконструкции.

Вид формул реконструкции и декомпозиции зависит от типа выбранных вейвлетов. В алгоритме, предложенном в настоящей работе, применяются сплайн-вейвлеты. Для сплайн-вейвлетного преобразования формулы декомпозиции имеют вид:

$$a_j = c_j, \quad j \le k-2;$$
$$a_j = c_{j+2}, \quad j \ge k-1;$$
$$b_{k-1} = c_{k-1} - \frac{x_{k+1} - x_k}{x_{k+1} - x_{k-1}} c_{k-2} - \frac{x_k - x_{k-1}}{x_{k+1} - x_{k-1}} c_k;$$
$$b_j = 0, \quad j \ne k-1,$$

а формулы реконструкции выражаются таким образом:

$$c_j = a_j, \quad j \le k-2;$$
$$c_j = a_{j-2}, \quad j \ge k-1;$$
$$c_k = \left(a_{k-1} - \frac{x_{k+1} - x_k}{x_{k+1} - x_{k-1}} a_{k-2} - b_{k-1} \right) \left(\frac{x_{k+1} - x_{k-1}}{x_k - x_{k-1}} \right).$$

Для вейвлет-преобразования необходима также сетка $X = \{x_j\}$, т. е. дискретное множество чисел. Удаляя из этой сетки узлы x_k, возможно отбрасывать не актуальные, с какой-либо точки зрения, элементы информационного потока.

Представленный алгоритм, генерирует S-блоки составляя их из бент-функций, являющихся результатом работы алгоритма генерации бент-функций с помощью вейвлет-преобразования [15]. Ключевым элементом алгоритма генерации S-блоков является ограничение степени нелинейности, накладываемое на бент-функции, полученные в ходе работы алгоритма [15]. Для формирования S-блока отбираются бент-функции степени нелинейности не выше заданной.

В алгоритме используются такие представления булевых функций, как таблица истинности и алгебраическая нормальная форма (АНФ), или полином Жегалкина.

Под таблицей истинности следует понимать таблицу, строка которой представляет собой двоичный вектор, являющийся конкатенацией аргумента $\mathbf{x} \in \mathbb{Z}_2^n$ ($\mathbb{Z}_2^n$ — множество двоичных векторов длины n) и соответствующего значения $f(\mathbf{x})$ булевой функции. Строки в таблице истинности идут в порядке возрастания аргумента.

Помимо таблицы истинности булева функция может быть однозначно задана своей АНФ:

$$f(\mathbf{x}) = a_0 \oplus \sum_{i=1}^{n} a_i x_i \oplus \sum_{1<i<j<n}^{n} a_{ij} x_i x_j \oplus \ldots \oplus a_{1,\ldots,n} x_1 \cdot \ldots \cdot x_n,$$

где $\oplus$ — сложение производится по XOR; коэффициенты $a \in \{0, 1\}$.

В ходе работы алгоритма также вычисляется функция «следа» tr: $GF(2^n) \to GF(2)$:

$$tr(x) = x + x^2 + x^4 + \ldots + x^{2^{n-1}},$$

где $CF(q)$ — конечное поле порядка q, а операции сложения и возведения в степень выполняются в конечном поле $GF(2^n)$.

Блок-схема процедуры Generate_bent_func_tables(), генерирующей таблицы истинности бент-функций степени нелинейности не выше заданной, приведена на рис. 1.

В процедуре Generate_bent_func_tables() использованы следующие вспомогательные функции:

— get_TTS: рассчитывает таблицу истинности по паре чисел (a, d) в соответствии с формулой $f(\mathbf{x}) = tr(ax^d)$;

— get_ANF: рассчитывает АНФ по таблице истинности;

— процедура, проверяющая наличие в АНФ слагаемых, степени больше заданной.

Заметим, что в предложенном алгоритме генерации S-блоков можно задействовать и отличный от описанного в [10] алгоритм генерации бент-функций. Использование другого способа получения бент-функций не повлияет качественно на генерируемые S-блоки, если бент-функции будут иметь вид $tr(ax^{2^{2k}-2^k+1})$: $GF(2^n) \to GF(2)$, где $a \in GF(2^n)\backslash\{x^3 | x \in GF(2^n)\}$, n не делится на три, k взаимно простое с n. Утверждение, что булевы функции указанного вида действительно являются бент-функциями, подтверждается в [8].

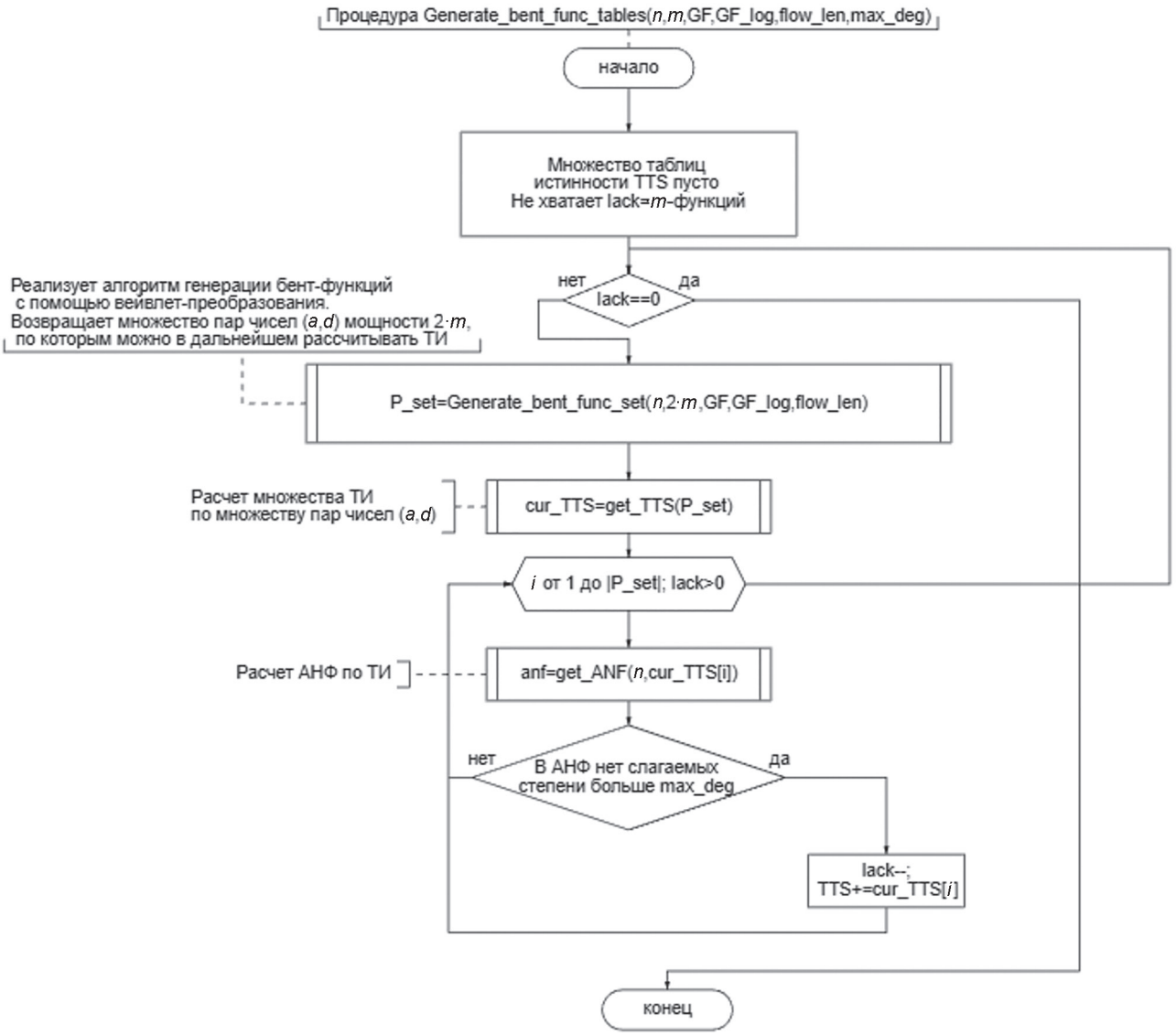

Рис. 1. Блок-схема процедуры Generate_bent_func_tables(), генерирующей таблицы истинности (ТИ) бент-функций

Fig. 1. Block diagram of the Generate_bent_func_tables() procedure, generating truth tables (ТИ) of bent functions

Алгоритм генерации S-блоков, блок-схема которого представлена на рис. 2, предполагает отбор блоков с характеристиками не хуже заданных, нелинейность получившегося блока должна быть не ниже заданной величины, а дифференциальная равномерность не выше. Входные данные этого алгоритма включают в себя входные данные для алгоритма генерации бент-функций с помощью вейвлет-преобразования, а также пороговые значения нелинейности, дифференциальной равномерности и степени нелинейности бент-функций, составляющих S-блок. При этом S-блоки являются векторами из бент-функций, сгенерированных при помощи процедуры Generate_bent_func_tables() (рис. 1).

Алгоритм генерации S-блоков был реализован на языке C++. Лучший из отобранных в процессе работы алгоритма S-блок, включен в состав хэш-функции «Стрибог», реализованной также в виде программы на C++. Программа не только реализовывала саму хэш-функцию, но и оценивала ее лавинный эффект.

Метод оценки лавинного эффекта хэш-функции «Стрибог»

Рассмотрим параметры, используемые для оценки лавинного эффекта, и способы их расчета.

Рассчитаем лавинный эффект хэш-функции на наборе входов **M** переменной длины

$$AE_R = \sum_{\mathbf{m}\in\mathbf{M}} \frac{hwt(hash(\mathbf{m}) \oplus hash(\mathbf{m} \oplus \mathbf{C}^r_{len(\mathbf{m})}))}{|\mathbf{M}|},$$

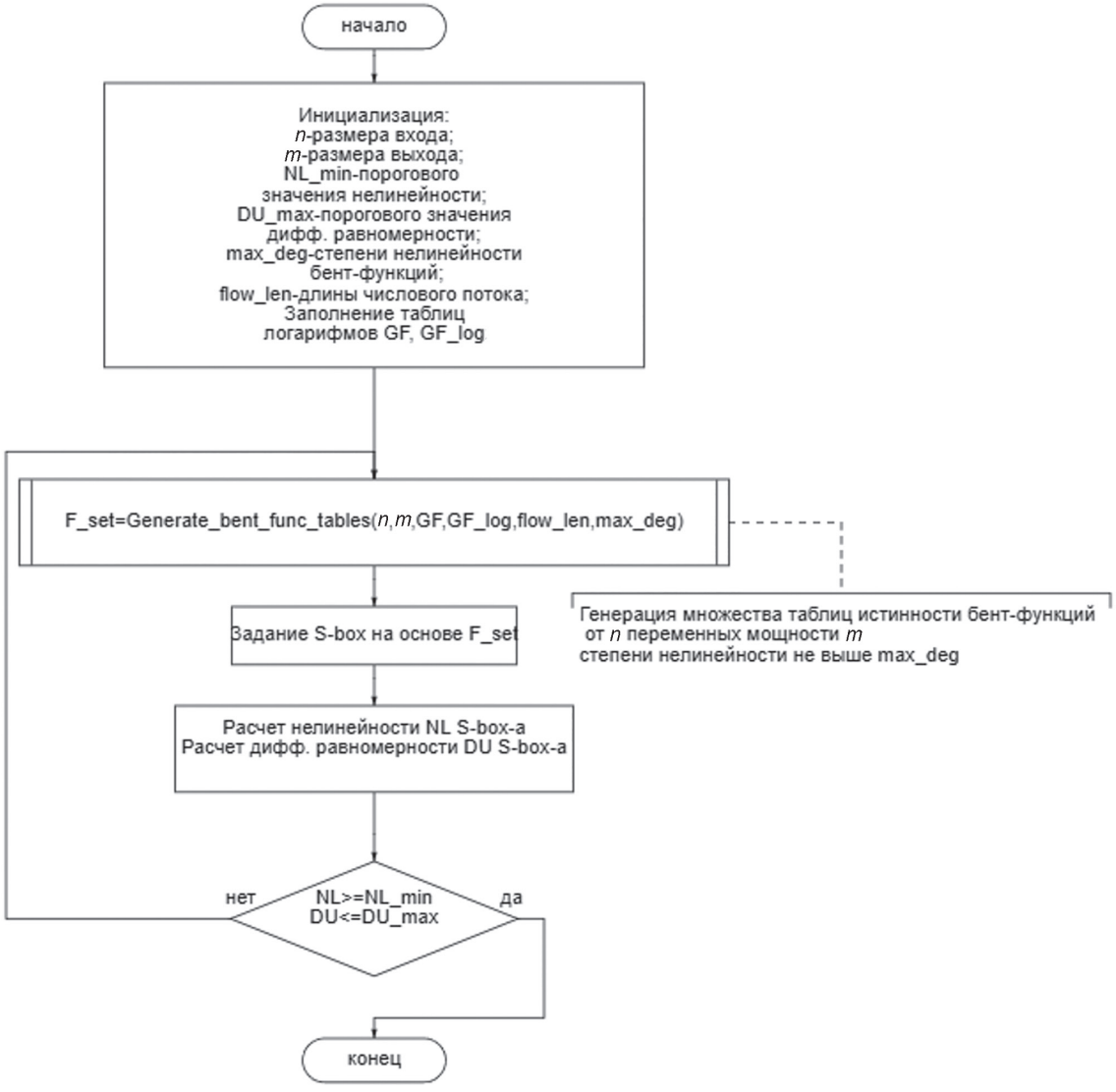

Рис. 2. Блок-схема предложенного алгоритма генерации S-блоков.

NL-min — минимальная допустимая нелинейность S-блока; DU-max — максимальная допустимая дифференциальная равномерность S-блока; NL S-box-a — нелинейность сгенерированного S-блока; DU S-box-a — дифференциальная равномерность сгенерированного S-блока

Fig. 2. Block diagram of the proposed algorithm generating S-boxes.

NL-min is the minimum permissible nonlinearity of the S-box; DU-max is the maximum permissible differential uniformity of the S-box; NL S-box-a is the nonlinearity of the generated S-box; DU S-box-a is the differential uniformity of the generated S-box

где $\forall \mathbf{m}$, $r \in U\,[0; len(\mathbf{m})]$ (при расчете для любого входа **m** параметр r выбирается случайно из интервала $[0; len(\mathbf{m})]$, где $len(\mathbf{m})$ — битовая длина **m**); $hwt(x)$ — вес Хэмминга битового вектора **x**, т. е. количество единиц в нем; $\mathbf{C}^{\mathbf{r}}_{len}(\mathbf{m})$ — битовый вектор длины $len(\mathbf{m})$, где все позиции, кроме r-й заняты нулями.

Для фиксированного битового размера l входных данных рассчитывалась аналогичная характеристика:

$$AE_F = \sum_{\mathbf{m}\in\mathbf{M}} \frac{\sum\limits_{0\le r\le l} hwt(hash(\mathbf{m}) \oplus hash(\mathbf{m} \oplus \mathbf{C}^{\mathbf{r}}_l))}{|\mathbf{M}|\cdot l},$$

где $\mathbf{C}^{\mathbf{r}}_l$ — битовый вектор длины l, при этом все позиции, кроме r-й заняты нулями.

Отметим, что в формуле расчета AE_F, параметр r «пробегает» все значения от 0 до l.

Из структурной схемы (рис. 3) видно, что ключевым элементом хэш-функции «Стрибог» является функция сжатия g.

Лавинный эффект был рассчитан для функции сжатия $g(\mathbf{h}, \mathbf{m}, \mathbf{N})$. В качестве входных данных рассматривались пары значений $(\mathbf{h}, \mathbf{m})$ из сформированного множества **M**, а **N** было равно 512 бит.

$$AE_g = \sum_{(\mathbf{h},\mathbf{m})\in\mathbf{M}} \frac{\sum\limits_{0\le r\le l} hwt(g(\mathbf{h}, \mathbf{m}) \oplus g(\mathbf{h}, \mathbf{m}) \oplus \mathbf{C}^{\mathbf{r}}_l))}{|\mathbf{M}|\cdot l},$$

где $l = 1024$ бит, так как $|\mathbf{h}| = |\mathbf{m}| = 512$ бит.

По аналогии с формулой AE_F, параметр r при расчете AE_g «пробегает» все значения от 0 до l.

Для функции сжатия $g(\mathbf{h}, \mathbf{m}, \mathbf{N})$ получена матрица лавинного эффекта SAC, выражающая вероятность изменения выходного бита с номером i при изменении входного бита j:

$$SAC_{i,j} = \frac{1}{|\mathbf{M}|} \sum_{(\mathbf{h},\mathbf{m})\in\mathbf{M}} g_i(\mathbf{h}, \mathbf{m}) \oplus g_i((\mathbf{h}, \mathbf{m}) \oplus \mathbf{C}^j_l),$$

где $\mathbf{C}^j_l$ — битовый вектор длины l, где все позиции, кроме j-й заняты нулями.

Матрица имеет размерность 1024 × 512 бит, что не позволяет отобразить ее наглядно, как в случае, например, с SAC матрицей S-блока, поэтому в настоящей работе в качестве показателя для оценки использовано среднее значение по матрице:

$$SAC = \frac{1}{|\mathbf{M}|\cdot l^2/2} \sum_{1\le i\le l/2} \sum_{1\le j\le l} \sum_{(\mathbf{h},\mathbf{m})\in\mathbf{M}} g_i(\mathbf{h}, \mathbf{m}) \oplus g_i((\mathbf{h}, \mathbf{m}) \oplus \mathbf{C}^j_l).$$

Отметим технические моменты оценки лавинного эффекта. Набор данных для расчета соответствующих показателей у хэш-функции с различными S-блоками в составе был одинаков.

В качестве входных данных использовались файлы со следующим содержанием и форматом записи: для расчета лавинного эффекта хэш-функции в целом использовались файлы с входами функции фиксированной или переменной длины (заполнялись такие файлы парами значений (длина случайной строки в байтах, случайная строка байт)); для расчета лавинного эффекта и SAC-функции сжатия использовались файлы, содержащие заданное количество пар $(\mathbf{h}, \mathbf{m})$, каждая пара — 1024 бита.

Важно было не только убедиться, что разработанные S-блоки не ухудшают лавинный эффект хэш-функции, но и оценить их преимущество в эффективности использования на ПЛИС.

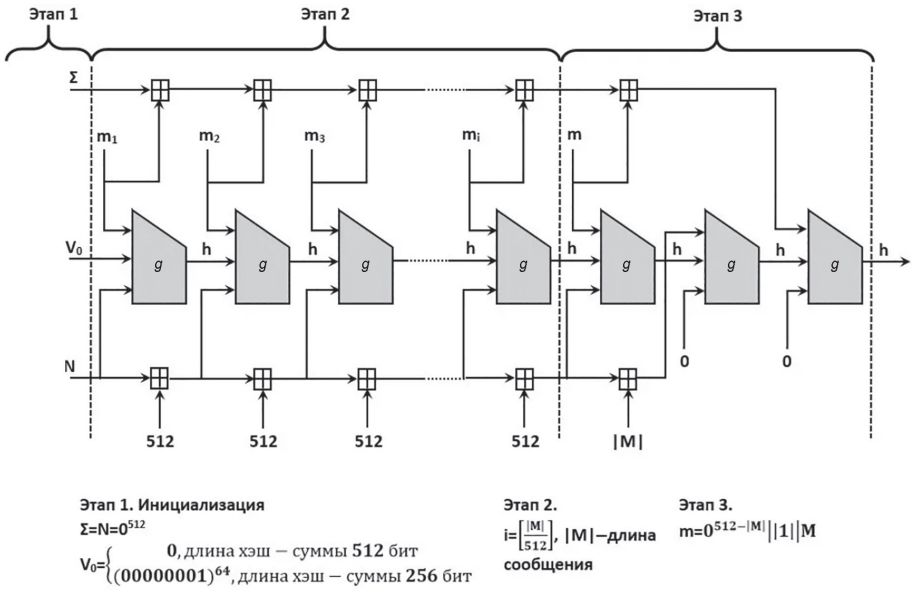

Рис. 3. Схема хэш-функции «Стрибог».

$\mathbf{m}_1$–$\mathbf{m}_i$ — блоки сообщения; **Σ** — контрольная сумма блоков; $\mathbf{V}_0$ — инициализирующий вектор; **N** — счетчик бит

Fig. 3. Streebog hash function scheme.

$\mathbf{m}_1$–$\mathbf{m}_i$ — message blocks; **Σ** — block checksum; $\mathbf{V}_0$ — initialization vector; **N** — bit counter

Основные результаты

Анализ результатов работы предложенного алгоритма генерации S-блоков. Основными и наиболее важными показателями, по которым различались сгенерированные алгоритмом S-блоки, являются нелинейность и дифференциальная равномерность.

Для проведения расчетов создавались S-блоки размерности 8 × 8 байт. Для всех значений степени нелинейности бент-функций, составляющих S-блок формировалось 100 различных блоков. Расчет характеристик производился программой на C++, а диаграммы строились с использованием Python библиотеки Mathplotlib.

Наиболее информативным является совместное распределение по дифференциальной равномерности и нелинейности. Распределение приводилось в виде столбчатой диаграммы. Лучшие результаты показали блоки, составленные из бент-функций степени нелинейности, не превышающей 2, их распределение приведено на рис. 4. Под лучшими результатами нужно понимать наибольшую долю блоков, сочетающих высокую нелинейность и низкую дифференциальную равномерность.

Сравнение характеристик построенных и существующих S-блоков. Сравнение лучших представителей разработанных S-блоков проводилось с S-блоками, используемыми в шифрах «Кузнечик» и AES, а также новыми улучшенными S-блоками AES [3].

В табл. 1 приведены численные значения следующих криптографических характеристик S-блоков: нелинейность, дифференциальная равномерность, лавинный эффект, максимальное значение корреляции в матрице BIC, DSAC (мера того, насколько S-блок далее от выполнения SAC).

Как видно из табл. 1, разработанный S-блок не уступает существующим, а по DSAC, BIC и лавинному эффекту превосходит их.

Заметим, что предложенные S-блоки не являются биективными, т. е. их невозможно применить в симметричных шифрах типа AES, «Кузнечик». Однако они могут быть с успехом использованы в тех примитивах, где нет необходимости иметь обратный S-блок, например, в шифрах, основанных на сети Фейстеля, хэш-функциях.

Лавинный эффект хэш-функции «Стрибог». В табл. 2 значения параметров, характеризующих лавинный эффект хэш-функции «Стрибог»: лавинный эффект функции в целом на наборе данных с произвольной длиной (AE_R); лавинный эффект функции в целом на наборе данных с постоянной длиной (AE_F); лавинный эффект функции сжатия (как видно из рис. 3, функция сжатия g — основной элемент хэш-функции «Стрибог») (AE_g); среднее значение по матрице SAC.

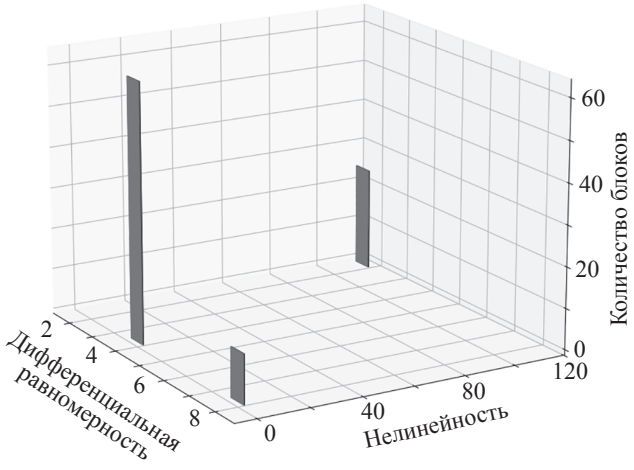

Рис. 4. Распределение S-блоков по нелинейности и дифференциальной равномерности (степень нелинейности функций не более 2)

Fig. 4. Distribution of S-boxes by nonlinearity and differential uniformity (degree of non-linearity equals 2)

Перечисленные параметры приводятся для хэш-функции «Стрибог» с блоком, используемым в настоящее время в соответствии с ГОСТ[1] (этот же S-блок применяется и в шифре «Кузнечик») и блоком, сгенерированным алгоритмом, рассмотренным в данной работе.

Расчеты соответствующих характеристик для разных блоков проводились на одинаковых наборах данных. Наборы данных имели ограниченный размер, так как перебор всех возможных входов для хэш-функции в целом невозможен в принципе, а для функции сжатия затруднителен.

Из данных табл. 2 можно сделать вывод об идентичности характеристик стандартного и предложенного S-блоков в части, касающейся влияния на лавинный эффект хэш-функции.

[1] ГОСТ Р 34.11–2012. Информационная технология. Криптографическая защита информации. Функция хэширования. Введ. 01.01.2013. М.: Стандартинформ, 2013. 19 с.

Таблица 1. Сравнение построенных и существующих S-блоков

Table 1. Comparison of constructed S-boxes with existing ones

Характеристика	S-блок AES	S-блок «Кузнечик»	Новый улучшенный S-блок AES [3]	Разработанный S-блок	Оптимальное значение
Нелинейность	112	100	112	112	120
Дифференциальная равномерность	4	8	4	2	2
DSAC	432	516	328	0	0
AE усредненный	1034	1049,5	1026	1024	1024
Максимальное значение в матрице BIC	0,13	0,29	0,12	0	0

Примечание: серым цветом выделены ячейки со значениями, совпадающими с оптимальными.

Таблица 2. Сравнение лавинного эффекта хэш-функции «Стрибог» для различных S-блоков
Table 2. Comparison of hash function Streebog avalanche effect for various S-boxes

Используемый S-блок	AE_R	AE_F	AE_g	*SAC*
Стандартный	0,499942	0,499424	0,437493	0,500162
Разработанный	0,499993	0,500496	0,434116	0,500103

Таблица 3. Использование ресурсов ПЛИС при задействовании стандартного и разработанного S-блоков
Table 3. Utilization of FPGA resources when using the standard and developed S-boxes

Тип ресурса	Использовано		Зафиксировано	Запрещено	Всего доступно	% использования	
	стандартный S-блок	разработанный S-блок				стандартный S-блок	разработанный S-блок
Секционные Look Up Table (LUT)	29 431	28 405	0	0	32 600	90,28	87,13
LUT как логика	29 431	28 405	0	0	32 600	90,28	87,13
LUT как память	0	0	0	0	9600	0,00	0,00
Секционные регистры	16 499	16 497	0	0	65 200	25,31	25,30
Регистр как триггер	15 475	15 473	0	0	65 200	25,73	23,73
Регистр как защелка	1024	1024	0	0	65 200	1,57	1,57
Мультиплексоры F7	512	2048	0	0	16 300	3,14	12,56
Мультиплексоры F8	0	768	0	0	8150	0,00	9,42

Примечание: серым фоном выделены ячейки с долей, задействованных Look Up Table и мультиплексоров.

Для того чтобы оценить сколько ресурсов конкретной ПЛИС задействует хэш-функция «Стрибог» с тем или иным S-блоком потребовалось синтезировать код, описывающий хэш-функцию «Стрибог» на языке System Verilog, с помощью IDE Vivado 2025.1. В настройках проекта System Verilog была выбрана ПЛИС семейства Xilinx Artix-7 xc7a50tcpg236-1.

Проводился синтез двух вариантов реализации хэш-функции «Стрибог». Первая реализация была стандартной, т. е. в ней использовался S-блок, описанный в ГОСТ на хэш-функцию «Стрибог». Для применения во второй реализации был отобран S-блок из числа сгенерированных алгоритмом, рассмотренным в настоящей работе. За исключением модуля, описывающего S-блок, реализации были идентичны.

В табл. 3 представлены результаты отчетов по задействованию логических элементов при синтезе реализаций хэш-функции со стандартным и разработанным S-блоком.

Реализация предложенного S-блока является менее ресурсозатратной, чем реализация стандартного S-блока, в большей мере задействуются встроенные мультиплексоры, используется на 3,15 % меньше LUT (в абсолютном выражении около 1000).

Причина большей эффективности разработанного S-блока в том, что он не биективен и фактически, представляется меньшим количеством различных значений, что не ухудшает его нелинейности, дифференциальной равномерности, лавинного эффекта и BIC.

1000 LUT довольно существенное значение для небольших ПЛИС, сэкономленный ресурс может позволить разместить при необходимости дополнительные модули. В некоторых случаях 1000 LUT может оказаться критически важной, так как при использовании предложенного S-блока не потребуется смена аппаратной платформы или сокращение функционала.

Заключение

Разработан алгоритм генерации S-блоков. Предложенный алгоритм генерации S-блоков продемонстрировал возможность создания компактных и криптографически стойких таблиц замен, ориентированных на эффективную реализацию в программируемых логических интегральных схемах (ПЛИС). Полученные результаты подтверждают, что использование бент-функций, в частности квадратичных, в S-блоках позволяет достичь высоких показателей нелинейности и дифференциальной равномерности, превосходящих, или сопоставимых с известными аналогами, такими как S-блоки Advanced Encryption Standard (AES) и «Кузнечик».

Важным наблюдением является тот факт, что предложенные S-блоки не являются биективными, что ограничивает их применение в симметричных шифрах, требующих обратимости преобразований (например, AES или «Кузнечик»). Однако, в таких примитивах как хэш-функции, или шифры на основе сети Фейстеля, где обратимость S-блока не требуется, их использование представляется перспективным.

Сравнение лавинного эффекта хэш-функции «Стрибог» с оригинальным и разработанным S-блоком показало практическую идентичность характеристик.

Это свидетельствует о том, что замена S-блока не ухудшает криптографические свойства функции, что является ключевым условием для ее практического применения.

Наиболее значимым практическим результатом является снижение ресурсозатрат при реализации хэш-функции на ПЛИС. Экономия около 1000 Look Up Table (3,15 %) на ПЛИС Xilinx Artix-7 может быть критически важной для ресурсо-ограниченных систем. Это открывает возможности для размещения дополнительных функциональных модулей без изменения аппаратной платформы.

Литература

1. Matsui M. Linear cryptanalysis method for DES cipher // Lecture Notes in Computer Science. 1994. V. 765. P. 386–397. doi: 10.1007/3-540-48285-7_33
2. Zyubina D.A., Tokareva N.N. Cryptographic properties of a simple S-box construction based on a Boolean function and a permutation // Applied Discrete Mathematics. Application. 2020. N 13. P. 41–43. doi: 10.17223/2226308x/13/13
3. Nitaj A., Susilo W., Tonien J. A New improved AES S-box with enhanced properties // Lecture Notes in Computer Science. 2020. V. 12248. P. 125–141. doi: 10.1007/978-3-030-55304-3_7
4. Cui J., Huang L., Zhong H., Chang C., Yang W. An improved AES S-box and its performance analysis // International Journal of Innovative Computing, Information and Control. 2011. V. 7. N 5 (A). P. 2291–2302.
5. Zahid A.H., Arshad M.J. An innovative design of substitution-boxes using cubic polynomial mapping. // Symmetry. 2019. V. 11. N 3. P. 437. doi: 10.3390/sym11030437
6. Rothaus O. On “bent” functions // Journal of Combinatorial Theory, Series A. 1976. V. 20. N 3. P. 300–305. doi: 10.1016/0097-3165(76)90024-8
7. Dillon J.F. A survey of bent functions // The NSA Technical Journal. 1972. Special Issue. P. 191–215.
8. McFarland R.L. A family of difference sets in non-cyclic groups // Journal of Combinatorial Theory, Series A. 1973. V. 15. N 1. P. 1–10. doi: 10.1016/0097-3165(73)90031-9
9. Carlet C. Boolean functions for cryptography and error-correcting codes // Boolean Models and Methods in Mathematics, Computer Science, and Engineering. 2007. P. 257–398.
10. Biham E., Shamir A. Differential cryptanalysis of DES-like cryptosystems // Journal of Cryptology. 1991. V. 4. N 1. P. 372. doi: 10.1007/bf00630563
11. Токарева Н.Н. Нелинейные булевы функции: бент-функции и их обобщения: Теоретические результаты. Saarbrucken, LAP LAMBERT Academic Publishing, 2011. 180 с.
12. Mar P.P., Latt K.M. New analysis methods on strict avalanche criterion of S-Boxes // International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering. 2008. V. 2. N 12. P. 899–903.
13. Webster A.F., Tavares S.E. On the design of S-Boxes // Lecture Notes in Computer Science. 1985. V. 218. P. 523–534. doi: 10.1007/3-540-39799-X_41
14. Detombe J., Tavares S. Constructing large cryptographically strong S-boxes // Lecture Notes in Computer Science. 1993. V. 718. P. 165–181. doi: 10.1007/3-540-57220-1_60
15. Шибакин И.В., Левина А.Б. Алгоритм генерации бент-функций с помощью вейвлет–преобразования // Безопасность информационных технологий. 2025. Т. 32. № 4. С. 106–121. doi: 10.26583/bit.2025.4.08
16. Демьянович Ю.К., Ходаковский В.А. Введение в теорию вэйвлетов: учебное пособие. СПб.: Издательство ПГУПС, 2007. 50 с.

References

1. Matsui M. Linear cryptanalysis method for DES cipher. *Lecture Notes in Computer Science*, 1994, vol. 765, pp. 386–397. doi: 10.1007/3-540-48285-7_33
2. Zyubina D.A., Tokareva N.N. Cryptographic properties of a simple S-box construction based on a Boolean function and a permutation. *Applied Discrete Mathematics. Application*, 2020, no. 13, pp. 41–43. doi: 10.17223/2226308x/13/13
3. Nitaj A., Susilo W., Tonien J. A New improved AES S-box with enhanced properties. *Lecture Notes in Computer Science*, 2020, vol. 12248, pp. 125–141. doi: 10.1007/978-3-030-55304-3_7
4. Cui J., Huang L., Zhong H., Chang C., Yang W. An improved AES S-box and its performance analysis. *International Journal of Innovative Computing, Information and Control*, 2011, vol. 7, no. 5 (a), pp. 2291–2302.
5. Zahid A.H., Arshad M.J. An innovative design of substitution-boxes using cubic polynomial mapping. *Symmetry*, 2019, vol. 11, no. 3, pp. 437. doi: 10.3390/sym11030437
6. Rothaus O. On “bent” functions. *Journal of Combinatorial Theory, Series A*, 1976, vol. 20, no. 3, pp. 300–305. doi: 10.1016/0097-3165(76)90024-8
7. Dillon J.F. A survey of bent functions. *The NSA Technical Journal*, 1972, special issue, pp. 191–215.
8. McFarland R.L. A family of difference sets in non-cyclic groups. *Journal of Combinatorial Theory, Series A*, 1973, vol. 15, no. 1, pp. 1–10. doi: 10.1016/0097-3165(73)90031-9
9. Carlet C. Boolean functions for cryptography and error-correcting codes. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, 2007, pp. 257–398.
10. Biham E., Shamir A. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 1991, vol. 4, no. 1, pp. 372. doi: 10.1007/bf00630563
11. Tokareva N.N. *Nonlinear Boolean Functions: Bent Functions and their Generalizations*. LAP LAMBERT Academic Publishing, 2011, 180 p. (in Russian)
12. Mar P.P., Latt K.M. New analysis methods on strict avalanche criterion of S-Boxes. *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 2008, vol. 2, no. 12, pp. 899–903.
13. Webster A.F., Tavares S.E. On the design of S-Boxes. *Lecture Notes in Computer Science*, 1985, vol. 218, pp. 523–534. doi: 10.1007/3-540-39799-X_41
14. Detombe J., Tavares S. Constructing large cryptographically strong S-boxes. *Lecture Notes in Computer Science*, 1993, vol. 718, pp. 165–181. doi: 10.1007/3-540-57220-1_60
15. Shibakin I.V., Levina A.B. Algorithm for generation of bent functions using wavelet transform. *It Security (Russia)*, 2025, vol. 32, no. 4, pp. 106–121. (in Russian). doi: 10.26583/bit.2025.4.08
16. Demyanovich Yu.K., Khodakovsky V.A. *Introduction to Wavelet Theory: A Teaching Medium*. St. Petersburg, PGUPS Publ., 2007, 50 p. (in Russian)

Авторы

Шибакин Илья Владиславович — техник, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197376, Российская Федерация, https://orcid.org/0009-0002-8692-7474, shibakin12@mail.ru

Левина Алла Борисовна — кандидат физико-математических наук, доцент, доцент, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197376, Российская Федерация, sc 56427692900, https://orcid.org/0000-0003-4421-2411, Alla_levina@mail.ru

Authors

Ilya V. Shibakin — Technician, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg, 197376, Russian Federation, https://orcid.org/0009-0002-8692-7474, shibakin12@mail.ru

Alla B. Levina — PhD (Physics & Mathematics), Associate Professor, Associate Professor, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg, 197376, Russian Federation, sc 56427692900, https://orcid.org/0000-0003-4421-2411, Alla_levina@mail.ru

Куприянов Михаил Степанович — доктор технических наук, профессор, руководитель перспективных направлений, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197376, Российская Федерация, SC 56785609900, https://orcid.org/0000-0003-4695-4507, mskupriyanov@etu.ru

Mikhail S. Kupriyanov — D.Sc., Professor, Head of Prospective Areas, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, 197376, Russian Federation, SC 56785609900, https://orcid.org/0000-0003-4695-4507, mskupriyanov@etu.ru

Статья поступила в редакцию 17.12.2025
Одобрена после рецензирования 24.04.2026
Принята к печати 24.05.2026

Received 17.12.2025
Approved after reviewing 24.04.2026
Accepted 24.05.2026