МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи

Прудников Вадим Александрович

СИНТЕЗ И ИССЛЕДОВАНИЕ ПСЕВДО-ДИНАМИЧЕСКИХ ПОДСТАНОВОК

Специальность 2.3.6 –
«Методы и системы защиты информации, информационная безопасность»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Румянцев Константин Евгеньевич

Таганрог – 2025

ОГЛАВЛЕНИЕ

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ

sbox – Substitution-box. Подстановка, sbox, замена

PRF – Pseudorandom function. Псевдослучайная функция

PRP – Pseudorandom permutation. Псевдослучайная перестановка

RFID – Radio-frequency identification. Радиочастотная идентификация

IoT – Internet of things. Интернет вещей

NIST – National Institute of Standards and Technology. Национальный институт стандартов и технологий

PD-sbox – Pseudo-Dynamic Substitution Box. Псевдо-динамическая подстановка

ARX – Add–rotate–XOR. Сложение по модулю слова, побитовый сдвиг, сложение по модулю 2

AEAD – Authenticated encryption with associated data. Аутентифицированное шифрование со связанными данными

SP-сеть – Substitution-Permutation (Подстановочно-перестановочная) сеть

CTR – Counter. Режим счетчика

DDT – Differential Distribution Table. Таблица переходов разностей

LAT – Linear Approximation Table. Таблица линейных аппроксимаций

ЭВМ – Электронная вычислительная машина

HPAC-SBOX – Hybrid Prediction and Adaptive Chaos sbox. Подстановка на базе гибридного предсказания и адаптивного хаоса

XNOR – Exclusive NOR. Исключающее ИЛИ-НЕ

CASCADA – Characteristic Automated Search of Cryptographic Algorithms for Distinguishing Attacks. Автоматизированный поиск криптографических характеристик для атак различителей

SMT – Satisfiability modulo theories. Задача выполнимости формул в теориях

SAT – Boolean satisfiability problem. Проблема логической выполнимости

AVX – Advanced Vector Extensions. Расширение системы команд x86

ВВЕДЕНИЕ

Актуальность темы исследования. Операции замены или фиксированные подстановки sbox являются основным нелинейным элементом для множества современных псевдослучайных функций (Pseudo Random Function – PRF) и псевдослучайных перестановок (Pseudo Random Permutation – PRP), которые являются базовым элементом логической защиты информации в информационных системах. Под термином «подстановка» (sbox, замена) подразумевается как взаимно однозначное, так и не взаимно однозначное преобразование m-битного сообщения на входе в n-битное сообщение на выходе, где n не всегда равно m.

Нелинейность подстановок напрямую влияет на сложность криптоанализа псевдослучайных функций и псевдослучайных перестановок, операции замены являются одним из наиболее ресурсоёмких элементов при программной или аппаратной реализации. Так как PRF и PRP строятся по итеративной схеме (для достижения заданного уровня суммарной нелинейности преобразования), от нелинейных свойств фиксированных подстановок напрямую зависит количество итераций, затрачиваемых вычислительных ресурсов, а также время обработки и потребляемая мощность.

К фиксированным операциям подстановки предъявляется около десятка требований, напрямую влияющих на криптографическую стойкость PRF и PRP. Для малоресурсных псевдослучайных функций и псевдослучайных перестановок остро встаёт проблема защиты от побочных каналов утечки секретной информации. Это накладывает дополнительные требования и ограничения на подстановки.

Актуальность исследований заключается в том, что проблема синтеза подстановок, удовлетворяющих широкому спектру взаимоисключающих требований, является базовой при синтезе эффективных PRF и PRP. Псевдослучайные функции, работающие в режиме счётчика с

аутентификацией Галуа (Authenticated Encryption with Associated Data – AEAD), в большинстве случаев способны заменить современные блочные шифры, например, в протоколах SSH и TLS, а также OpenVPN, так как обладают более высокой производительностью и/или меньшим потреблением ресурсов при программной реализации. Однако применение псевдо-динамических подстановок при синтезе PRF потенциально способно обеспечить устойчивость к криптоанализу превосходящую аналоги, при сохранении сопоставимых затратах ресурсов при программной реализации. Следовательно, синтез и исследование псевдо-динамических подстановок является актуальной темой и представляет научный и практический интерес.

Степень разработанности темы. Существует множество подходов решения проблемы синтеза операций подстановки. Большинство заключается в применении различных методик при генерации фиксированных блоков замен, обладающих требуемыми криптографическими свойствами. Например, коллективом авторов – Ivanov G., Nikolov N., Nikova S. описан реверсивный генетический алгоритм, использование которого позволяет быстро генерировать большое число стойких биективных подстановок размерностью от 8 до 16 бит, которые имеют неоптимальные свойства и сложную алгебраическую структуру, а также не обладают линейной избыточностью. Автором Tesař P. представлен метод генерации таблиц подстановок размерностью 8 бит с нелинейностью, достигающей значения 104. Метод комбинирует специальный генетический алгоритм с полным деревом поиска. Коллективом авторов – Kazymyrov O., Kazymyrova V., Oliynykov R. представлен метод генерации нелинейных sbox на основе градиентного спуска. Использование предложенного метода для наиболее часто применяемых подстановок, размерностью 8 бит, позволяет добиться показателей нелинейности 104.

Указанные подходы не удовлетворяют всем взаимоисключающим требованиям. В частности, размерность сгенерированной подстановки может не позволить эффективно применять её в программной или аппаратной

реализации криптографических преобразований, в силу потребления большого объёма ресурсов.

Иной способ решения заключается в применении в качестве фиксированных подстановок ARX-функций. Например, авторами Bernstein D.J., Robshaw M., Billet O. представлено семейство поточных шифров Salsa20, основанное на ARX-операциях. Авторами Maitra S., Paul G., Meier W. представлены результаты криптоанализа Salsa20. Им удалось достичь сложности поиска ключа в $2^{247,2}$ при осуществлении анализа 8-раундовой реализации, что значительно превосходит результаты прошлых лет в 2^{251} и 2^{250}. Авторы Beaulieu R., Shors D., Smith J. разработали шифры Simon и Speck – легковесные блочные криптоалгоритмы, предназначенные для интернета вещей и построенные на основе ARX-функций. В статье «Cryptanalysis of the Speck Family of Block Ciphers» коллективом авторов – Abed F., List E., Lucks S., Wenzel J., представлены результаты дифференциального криптоанализа над описанными шифрами. Коллективом авторов (Beierle C., Micciancio D., Ristenpart T.) представлена 64-битная операция подстановки Alzette, основой которой являются ARX-функции. Особенностью преобразования является то, что оно вычисляется на современных процессорах за фиксированное время и использует всего 12 инструкций.

Недостатком подхода, подразумевающего использование ARX-операций, являются неудовлетворительные криптографические свойства создаваемых конструкций, однако они позволяют добиться высокого быстродействия и малого потребления ресурсов при программной и аппаратной реализации криптографических преобразований.

Для противодействия статистическим методам криптоанализа неоднократно осуществлялись попытки применять вместо фиксированных подстановок динамически изменяемые подстановки. Наиболее успешной попыткой применения динамически изменяемой подстановки можно назвать криптоалгоритм RC4, представленный автором Weerasinghe T. D. B. в работе «An Effective RC4 Stream Cipher», который считается устаревшим и

ненадёжным. Основная проблема стойкости RC4 – применение всего одной динамически изменяемой подстановки и медленное обновление содержимого подстановки (за одну итерацию обновляется 2 ячейки из 256), что опубликовано в статье Klein A. «Attacks on the RC4 stream cipher». Проблема предопределена тем, что динамические подстановки (в сравнении с фиксированными подстановками) требуют на порядки больше вычислительных ресурсов.

Научным коллективом авторов (Поликарпов С.В., Румянцев К.Е., Кожевников А.А., Петров Д.А.) предложен новый класс операций подстановки – псевдо-динамические подстановки (PD-sbox). Псевдо-динамические подстановки обладают как свойствами фиксированных подстановок (относительно низкие затраты вычислительных ресурсов), так и свойствами динамических подстановок (эффективное противодействие статистическим методам криптоанализа). Применение псевдо-динамических операций подстановки на базе фиксированных замен потенциально позволяет решить ряд описанных выше проблем, в частности обеспечить устойчивость к статистическим методам криптоанализа. В свою очередь, применение подобранных ARX-функций для использования в структуре псевдо-динамических подстановок потенциально позволяет получить вес дифференциальных и линейных характеристик, превосходящий аналоги, при тех же затратах ресурсов при программной реализации криптографических преобразований.

В связи с вышесказанным возникает актуальная научная задача разработки и исследования метода синтеза псевдо-динамических подстановок на основе ARX-функций, удовлетворяющих широкому спектру противоречивых требований по стойкости к криптоанализу и затрачиваемым ресурсам при программной реализации криптографических преобразований.

Целью диссертационного исследования является минимизация затрачиваемых ресурсов программной реализации криптографических преобразований при обеспечении заданных криптографических свойств

посредством разработки метода синтеза псевдо-динамических подстановок на основе ARX-функций.

Достижение поставленной цели предусматривает решение **частных задач:**

1. Анализ существующих подходов к синтезу псевдо-динамических операций подстановки.

2. Синтез структуры псевдо-динамической операции подстановки, удовлетворяющей широкому спектру противоречивых требований, посредством разработки метода синтеза псевдо-динамических подстановок на основе ARX-функций.

3. Анализ синтезированной псевдо-динамической функции PD-sbox-ARX-32 и её программной реализации на малоресурсных процессорах.

Объект исследования – криптографические операции подстановки, являющиеся составным элементом множества блочных шифров.

Предмет исследования – синтез и исследование псевдо-динамических операций подстановки, удовлетворяющих широкому спектру противоречивых требований по стойкости к разностному криптоанализу, а также затрачиваемым ресурсам при программной реализации криптографических преобразований.

Методы исследования: статистический криптоанализ с использованием SMT/SAT решателей, численные методы для оценки свойств псевдо-динамических подстановок, вычислительный эксперимент по определению криптографических свойств ARX-функций и псевдо-динамической функции PD-sbox-ARX-32.

Основные научные положения, выносимые на защиту:

1. Существующие подходы к синтезу и применению динамических операций подстановки не позволяют одновременно обеспечить стойкость, минимизацию затрачиваемых ресурсов и скорость программной реализации псевдослучайных функций на их основе, сопоставимую с псевдослучайными функциями на основе фиксированных подстановок или иных фиксированных

преобразований. В отличие от этого, метод синтеза псевдо-динамических подстановок на основе ARX-функций позволяет получать преобразования, удовлетворяющие требованиям по криптографическим свойствам, затрачиваемым ресурсам и скорости программной реализации криптографических преобразований.

2. Синтезированная структура 32-битной ARX-функции в составе PD-sbox позволяет обеспечить критический путь (максимальное количество последовательных операций сложения по модулю 2^{16}) в четыре раза меньше, чем ARX-преобразования, такие как 8-итерационная 32-битная Alzette-подобная структура, или 8-итерационное 32-битное преобразование криптоалгоритма Speck32, при двукратном увеличении количества операций и при сопоставимых максимальных значениях весов разностных и линейных характеристик.

3. Разработанный метод синтеза PD-sbox-ARX позволяет путём подбора параметров ARX-функций минимизировать количество затрачиваемых ассемблерных инструкций на операции циклического сдвига при их реализации на малоресурсных 8-битных микроконтроллерах семейства AVR (например, ATmega328P) и, в отличие от метода случайного поиска оптимальных параметров, позволяет снизить количество соответствующих ассемблерных инструкций на 23,6% при программной реализации псевдо-динамической подстановки, включающей в свой состав четыре 32-битные ARX-функции.

4. Разработанный метод синтеза псевдо-динамических подстановок на основе ARX-функций позволяет подобрать параметры для 32-битных ARX-функций, при которых, в отличие от 8-итерационного 32-битного преобразования криптоалгоритма Speck32, требуется на 10,6% меньше ассемблерных инструкций на операции циклического сдвига при их реализации на малоресурсных 8-битных микроконтроллерах семейства AVR, и обеспечивается максимальный вес разностной характеристики, равный 2^{-32} (эмпирический вес 2^{-26}), и вес линейной характеристики 2^{-13}.

Научная новизна состоит в следующем:

1. Разработана структура псевдо-динамической операции подстановки на основе ARX-функций, обладающая свойствами эквивалентных замен, аналогичными случайно сформированным операциям подстановки той же размерности (пункт 19 паспорта специальности).

2. Разработан и исследован метод синтеза параметров 32-битной ARX-функции, позволяющий получить параметры операций циклического сдвига, при которых обеспечивается максимальный вес разностной характеристики, равный 2^{-32} (эмпирический вес 2^{-26}), и вес линейной характеристики 2^{-13} для результирующего PD-sbox-ARX, включающей в свой состав четыре 32-битные ARX-функции, а также позволяющий минимизировать количество затрачиваемых ассемблерных инструкций на операции циклического сдвига при реализации на малоресурсных 8-битных микроконтроллерах семейства AVR (пункт 19 паспорта специальности).

Теоретическая значимость результатов исследования состоит в развитии перспективного научного направления синтеза и применения псевдо-динамических подстановок на основе ARX-функций, удовлетворяющих широкому спектру противоречивых требований по стойкости к криптоанализу и затрачиваемым ресурсам при программной реализации криптографических преобразований.

Практическая ценность работы:

Применение псевдо-динамических подстановок на базе подобранных ARX-функций, обладающих дифференциальными и линейными свойствами эквивалентных подстановок, аналогичными случайно сформированным фиксированным подстановкам той же размерности, позволяет обеспечить критический путь (максимальное количество последовательных операций сложения по модулю 2^{16}) в четыре раза меньше, чем 8-итерационная 32-битная Alzette-подобная структура, при двухкратном увеличении количества операций и при сопоставимых максимальных значениях весов разностных и линейных характеристик. При аппаратной реализации ARX-функции данное

свойство позволяет пропорционально уменьшить (до 4 раз) задержку при преобразовании блоков информации.

Достоверность результатов диссертационной работы подтверждается сходимостью исходной гипотезы с результатами опытно-экспериментальных данных, а также строгостью применяемого математического аппарата.

Внедрение результатов работы. Результаты диссертационного исследования, подтверждённые соответствующими актами, используются в:

при подаче заявки «Метод синхронизации между абонентами локальной квантовой сети и доверенным узлом магистральной квантовой сети» на конкурс 2024 года Российского научного фонда «Проведение фундаментальных научных исследований и поисковых научных исследований малыми отдельными научными группами». В частности, формулирование частной научной задачи гранта, связанной с поиском возможных контрмер против выявленных атак, основывается на научных результатах диссертационной работы в части метода синтеза псевдо-динамической подстановки PD-sbox-ARX-32 и структуры псевдо-динамической операции подстановки на основе ARX-функций;

научной деятельности кафедры Информационной безопасности телекоммуникационных систем Института компьютерных технологий и информационной безопасности Южного федерального университета;

учебном процессе кафедры Информационной безопасности телекоммуникационных систем Института компьютерных технологий и информационной безопасности Южного федерального университета, в части разработанной программы для ЭВМ.

Апробация результатов. Основные результаты работы докладывались и обсуждались на 5 научных конференциях:

I Всероссийская научно-практическая конференция «Digital Era», г. Грозный, 26 марта 2021;

VII Всероссийская научно-техническая конференция «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности», г. Таганрог, 05-11 апреля 2021;

VIII Всероссийская научно-техническая конференция «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности», г. Таганрог, 04-09 апреля 2022;

IX Всероссийская научно-техническая конференция «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности», г. Таганрог, 10-15 апреля 2023;

XII симпозиум «Современные тенденции в криптографии» (CTCrypt 2023) 6-9 июня 2023, г. Волгоград.

Публикации. Основные положения диссертации опубликованы в 11 научных печатных работах, в том числе: 5 – в ведущих рецензируемых научных журналах, входящих в перечень ВАК РФ (из них 1 категории К1 и RSCI, 4 категории К2), 6 – в материалах конференций и других изданиях. Получено свидетельство о государственной регистрации программы для ЭВМ.

Соответствие паспорту специальности. Диссертация соответствует пункту 19 «Исследования в области безопасности криптографических алгоритмов, криптографических примитивов, криптографических протоколов. Защита инфраструктуры обеспечения применения криптографических методов» паспорта научной специальности 2.3.6 – «Методы и системы защиты информации, информационная безопасность».

Личный вклад автора. Основные научные результаты, в том числе структура псевдо-динамической операции подстановки на основе ARX-функций, метод синтеза параметров 32-битной ARX-функции, а также количественная оценка затрачиваемых ресурсов программной реализации разработанного криптографического преобразования и криптографических свойств получены автором лично.

Структура и объем диссертации. Диссертация написана на русском языке, состоит из введения, трёх глав, заключения, списка используемых

источников из 71 наименования и приложения. Полный объём диссертации составляет 133 страницы (в том числе приложения – 7 страниц), включая 31 рисунок, 28 таблиц.

Во введении обосновывается актуальность темы, формулируются научная задача исследования, определяются объект и предмет исследования, практическая ценность и научная новизна результатов, излагаются научные положения, выдвигаемые на защиту.

В первой главе содержится анализ существующих подходов к синтезу псевдо-динамических операций подстановки. Приводится анализ синтеза операций подстановки, как основного нелинейного элемента современных блочных шифров и псевдослучайных функций. Даётся описание структуры псевдо-динамической операции подстановки PD-sbox, линейный и дифференциальный криптоанализ PD-sbox на основе фиксированных операций подстановки. Представлено описание метода автоматизированного поиска криптографических характеристик с использованием SMT решателей и библиотеки CASCADA. Приведены выводы о том, что существующие подходы к синтезу и применению динамических операций подстановки не позволяют одновременно обеспечить стойкость, минимизацию затрачиваемых ресурсов и скорость программной реализации псевдослучайных функций на их основе, сопоставимую с псевдослучайными функциями на основе фиксированных подстановок. В связи с этим синтез псевдо-динамических подстановок, удовлетворяющих взаимоисключающим требованиями, в частности дифференциальным характеристикам, не уступающим фиксированным операциям подстановки той же размерности, является актуальной проблемой. Результатом является постановка общей научной задачи и формулировка частных задач диссертационных исследований.

Во второй главе содержится описание синтеза структуры псевдо-динамической операции подстановки на основе ARX-функций. Исследования демонстрируют, что объединение ARX-функций, имеющих откровенно слабые криптографические свойства, в структуру псевдо-динамической

подстановки позволяет получать свойства эквивалентных подстановок, близкие к свойствам случайно сформированных подстановок аналогичной размерности. PD-sbox-ARX содержит простые операции и имеет заложенные возможности параллелизации обработки данных, что позволяет делать эффективные программные и аппаратные реализации для различных процессоров и аппаратных платформ.

Предложен метод синтеза псевдо-динамической функции PD-sbox-ARX-32, который позволяет получать PD-sbox-ARX с достаточно близкими к 8-раундовым преобразованиям Speck32 и miniAlzette32 криптографическими свойствами. При синтезе 100 PD-sbox-ARX 73 варианта имели вес разностных характеристик Wd равный 32 и вес линейных характеристик Wl, равный 13 и 14.

В третьей главе приведены результаты исследования дифференциальных и линейных характеристик PRF pCollapserARX, используя CASCADA. Проанализирован метод синтеза PD-sbox-ARX. Сделаны выводы о том, что подобранная структура 32-битной ARX-функции в составе PD-sbox позволяет обеспечить критический путь (максимальное количество последовательных операций сложения по модулю 2^{16}) в четыре раза меньше, чем 8-итерационная 32-битная Alzette-подобная структура, при двухкратном увеличении количества операций и при сопоставимых максимальных значениях весов разностных и линейных характеристик.

Аналогичный результат получается при сравнении 32-битной ARX-функции с 8-итерационным 32-битным преобразованием из блочного криптоалгоритма Speck32. При аппаратной реализации ARX-функции данное свойство позволяет пропорционально уменьшить (до 4 раз) задержку при преобразовании блоков информации.

Предложенный метод синтеза параметров 32-битной ARX-функции позволяет получить параметры операций циклического сдвига, при которых обеспечивается максимальный вес разностной характеристики равный 2^{-32} (эмпирический вес 2^{-26}) и вес линейной характеристики 2^{-13} для

результирующего PD-sbox-ARX, включающей в свой состав четыре 32-битные ARX-функции. Сопоставимые разностные и линейные характеристики имеют 8-итерационные 32-битная Alzette-подобная структура и 8-итерационное 32-битное преобразование из блочного криптоалгоритма Speck32.

Предложенный метод синтеза параметров 32-битной ARX-функции позволяет минимизировать количество затрачиваемых ассемблерных инструкций на операции циклического сдвига при реализации на малоресурсных 8-битных микроконтроллерах семейства AVR (например, ATmega328P).

В заключении формулируются выводы, основные результаты работы и рекомендации.

В приложениях приводятся акты о внедрении результатов диссертационной работы, а также свидетельство о государственной регистрации программы для ЭВМ.

1. АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ К СИНТЕЗУ ПСЕВДО-ДИНАМИЧЕСКИХ ОПЕРАЦИЙ ПОДСТАНОВКИ

1.1. Анализ операций подстановки sbox как основного нелинейного элемента современных блочных шифров и псевдослучайных функций

Операции замены или фиксированные подстановки sbox являются основным нелинейным элементом для множества современных псевдослучайных функций PRF и псевдослучайных перестановок PRP [1–4], которые являются базовым элементом логической защиты информации в информационных системах.

Под термином «подстановка» (sbox, замена) в работе подразумевается, как взаимнооднозначное, так и невзаимнооднозначное преобразование m-битного сообщения на входе в n-битное сообщение на выходе, где n не всегда равно m.

Нелинейность подстановок напрямую влияет на сложность криптоанализа PRF и PRP, операции замены являются одним из наиболее ресурсоёмких элементов при программной или аппаратной реализации. Так как PRF и PRP строятся по итеративной схеме (для достижения заданного уровня суммарной нелинейности преобразования), то от нелинейных свойств фиксированных подстановок напрямую зависит количество итераций, затрачиваемых вычислительных ресурсов, а также время обработки и потребляемая мощность.

Активно развиваются малоресурсные (легковесные) псевдослучайные функции и псевдослучайные перестановки [1–3], для которых в ущерб стойкости и времени обработки информации улучшаются показатели по затрачиваемым вычислительным ресурсам и потребляемой мощности. Их появление предопределено внедрением защищённых RFID меток, смарт-карт, устройств IoT и электронных устройств с ограниченными аппаратными

ресурсами. На текущий момент представлено более 50 легковесных блочных криптоалгоритмов, десятки поточных криптоалгоритмов и хеш-функций. Часть из них является международными или национальными стандартами [5–7]. Институт NIST США запустил конкурс по отбору и стандартизации легковесных псевдослучайных функций [8] для малоресурсной электроники.

К фиксированным операциям подстановки предъявляется около десятка требований, напрямую влияющих на криптографическую стойкость PRF и PRP [9]. Для малоресурсных псевдослучайных функций и псевдослучайных перестановок остро встаёт проблема защиты от побочных каналов утечки секретной информации. Это накладывает дополнительные требования и ограничения на подстановки.

Исследованию свойств и синтезу фиксированных подстановок научное сообщество посвятило более 40 лет (с момента появления первых блочных шифров). Научным коллективом авторов (Поликарпов С.В., Румянцев К.Е., Кожевников А.А., Петров Д.А.) предложен новый класс операций подстановки – псевдо-динамические подстановки (PD-sbox). Псевдо-динамические подстановки обладают как свойствами фиксированных подстановок (относительно низкие затраты вычислительных ресурсов), так и свойствами динамических подстановок (эффективное противодействие статистическим методам криптоанализа). Уникальные свойства псевдо-динамических подстановок требуют детальных исследований их криптографических характеристик.

Блок криптографической подстановки (sbox) – это элемент, осуществляющий отображение n-битного сообщения на входе в m-битное сообщение на выходе. Блоки замен обладают множеством криптографических свойств: нелинейность; дифференциальные характеристики; сбалансированность; корреляционный иммунитет; глобальный лавинный критерий; алгебраический иммунитет; критерий распространения; порядок прозрачности.

Обозначенные параметры криптографических подстановок оказывают ключевое влияние на устойчивость криптоалгоритмов и псевдослучайных функций к различным методам криптоанализа.

Криптографические операции подстановки sbox являются основным нелинейным элементом множества современных блочных шифров и псевдослучайных функций. Их устойчивость к различным методам криптоанализа напрямую зависит от типа и качества используемых операций подстановки.

Одной из основных задач блока замены является обеспечение устойчивости к статистическим методам криптоанализа, в частности к линейному и дифференциальному. Подбор операций подстановки для криптоалгоритмов или псевдослучайных функций не является тривиальной задачей, основная проблема – анализ множества синтезируемых криптографических подстановок для отбора структур, соответствующих взаимоисключающим критериям, которые определяют элемент, максимально приближенный к идеальному. При генерации подстановок необходимо соблюдать множество жёстких требований для обеспечения стойкости к статистическим атакам. Синтез криптоустойчивых замен необходим как для разрабатываемых алгоритмов, так и для использующихся в настоящее время.

Проблема синтеза операций подстановки, удовлетворяющих широкому спектру взаимоисключающих требований по устойчивости к различным методам криптоанализа и потреблению как программных, так и аппаратных ресурсов, является актуальной и ей уделяется значительное внимание. Существует множество подходов к решению проблемы синтеза операций подстановки. Большинство заключается в применении различных методик при генерации фиксированных блоков замен, обладающих требуемыми криптографическими свойствами. Иным способом решения задачи является применение конструкций, потенциально способных заменить криптографические операции подстановки в шифрах и псевдослучайных функциях, к ним относятся ARX-конструкции (структуры, включающие в

свой состав операции сложения по модулю слова, циклического сдвига и XOR), динамические операции подстановки, позволяющие потенциально нивелировать возможность применения статистических атак на криптоалгоритм, псевдо-динамические операции подстановки, включающие в свой состав либо фиксированный замены, либо специально подобранные ARX-функции, и позволяющие объединить преимущества как классических sbox, так и динамических, что даёт ряд преимуществ при их аппаратной и программной реализации в составе псевдослучайных функций [10].

Проанализируем первый вариант решения проблемы – синтез криптографических операций подстановки с использованием различных алгоритмов, позволяющих получить элемент, обладающий криптографическими свойствами, приближенными к идеальным.

В [9] описан реверсивный генетический алгоритм, использование которого позволяет быстро генерировать большое число стойких биективных подстановок размерностью от 8 бит до 16, которые имеют неоптимальные свойства и более сложную алгебраическую структуру, а также не обладают линейной избыточностью.

В [11] представлен метод генерации sbox размерностью 8 бит с нелинейностью, достигающей значения 104. Метод комбинирует специальный генетический алгоритм с полным деревом поиска.

В [12] представлен метод генерации нелинейных sbox на основе градиентного спуска. Приведены критерии отбора операций подстановки для криптографических симметричных примитивов, основанных на анализе свойств векторных булевых функций. Предлагается усовершенствованный метод градиентного спуска для увеличения эффективности генерации нелинейных векторных булевых функций с оптимальными криптографическими показателями. Использование предложенного метода для наиболее часто применяемых подстановок, размерностью 8 бит, позволяет добиться показателей нелинейности 104.

Авторами работы [13] предлагается подход к генерации операций подстановки, основанный на применении четвертичных последовательностей де Брейна, позволяющих добиться значительного увеличения числа доступных экономичных sbox по сравнению с использованием двоичных последовательностей де Брейна.

Исследования в [14] посвящены разработке новой реализации криптоалгоритма AES, включающего в свой состав HPAC-SBOX (Hybrid Prediction and Adaptive Chaos – гибридное прогнозирование и адаптивный хаос), который объединяет алгоритмы обучения с прогнозированием и адаптивные хаотические операции подстановки sbox.

В [15] сравнивается эффективность подходов к генерации операций подстановки в соответствии с их значениями нелинейности. Рассмотрены преимущества и недостатки представленных подходов.

В [16] представлена разработка алгоритма генерации sbox с использованием генетического алгоритма. В алгоритме генерации обработано значение нелинейности, которое является одним из наиболее важных критериев оценки операций подстановки. Качество сгенерированных блоков замен определено с помощью тестов производительности.

В [17] предлагается алгоритм генерации операций подстановки, основанный на 4D гиперхаотической системе и улучшенной оптимизации роя частиц. Улучшена хаотическая система Лоренца и предложена 4D гиперхаотическая система с более высоким показателем Ляпунова и более сложной динамикой. Идея алгоритма имитационного отжига введена в алгоритм оптимизации роя частиц, что повышает эффективность алгоритма оптимизации и устраняет проблему, заключающуюся в том, что алгоритм оптимизации роя частиц легко поддается локальному оптимальному решению. Алгоритм использован для оптимизации нелинейности блоков замен и повышения производительности sbox.

Указанные подходы не удовлетворяют всем взаимоисключающим требованиям. В частности, размерность сгенерированной подстановки может

не позволить эффективно применять её в программной или аппаратной реализации в силу потребления большого объёма ресурсов.

Иной способ решения заключается в применении в качестве фиксированных подстановок ARX-функций.

В [18] представлено семейство поточных шифров Salsa20, основанное на ARX-операциях. Классическая версия криптоалгоритма включает 20 раундов преобразований и три вида операций над 32-битными словами: сложение по модулю 2^{32}, операция XOR, циклический сдвиг. Salsa20 расширяет 256-битный ключ и 64-битный nonce (уникальный номер сообщения) в 270-байтовый поток. Он шифрует b-байтовый открытый текст, объединяя открытый текст с первыми b байтами потока и отбрасывая остальную часть потока. Операция дешифрования осуществляется выполнением операции XOR над зашифрованным текстом с первыми b байтами потока. В алгоритме отсутствует обратная связь от открытого или зашифрованного текста к потоку. Salsa20 генерирует поток блоками по 64 байта (512 бит). Каждый блок включает независимый хэш ключа, nonce и 64-битный номер блока, отсутствует сцепление предыдущего блока с последующим. Поток на выходе криптоалгоритма может быть доступен случайным образом, и любое количество блоков может быть вычислено параллельно. В Salsa20 нет предварительной обработки. В [19] представлены результаты криптоанализа над семейством поточных шифров Salsa20. Авторам удалось достичь сложности поиска ключа в $2^{247,2}$ при осуществлении анализа 8-раундовой реализации, что значительно превосходит результаты прошлых лет в 2^{251} и 2^{250}.

Работа [20] посвящена новому методу поиска линейных аппроксимаций криптоалгоритмов на основе ARX-конструкций, в частности шифра ChaCha. Авторами демонстрируется получение линейных аппроксимаций для 3 и 4 раундов ChaCha. В [21] представлены улучшения в системе дифференциально-линейных атак, предназначенных для шифров на базе ARX-операций. Для

демонстрации результатов работы применены к криптоалгоритмам Chaskey и ChaCha.

В [22] представлены шифры Simon и Speck – легковесные блочные криптоалгоритмы, предназначенные для интернета вещей. Максимальный размер блока составляет 128 бит, максимальный размер ключа – 256 бит. Блок состоит из двух слов, при этом слово может иметь размер 16, 24, 32, 48 или 64 бит. Ключ обладает размерностью в 2, 3 или 4 слова. Раундовая функция включает в себя операции: циклического сдвига первого слова вправо на 8 бит, сложение второго слова с первым по модулю 2 в степени длины слова, операция XOR ключа и результата сложения, циклический сдвиг второго слова влево на 3 бита, операция XOR второго слова и результата предыдущего XOR. Количество раундов зависит от выбранных размеров слова и ключа, для максимальных размеров блока и ключа количество раундов равно 34, при минимальных значениях – 22. В [23] представлены результаты дифференциального криптоанализа над описанными шифрами. В [24] представлен новый блочный шифр на базе ARX-конструкций и MDS-матрицы на основе концепции белого ящика – WARX.

В [25] представлена 64-битная операция подстановки Alzette, основой которой являются ARX-функции. Особенностью преобразования является то, что оно вычисляется на современных процессорах за фиксированное время и использует всего 12 инструкций. Параллельная реализация Alzette может использовать векторные (SIMD) инструкции. Одна итерация обладает дифференциальными и линейными характеристиками, сравнимыми со свойствами операции подстановки алгоритма AES, две последующие итерации обеспечивают тот же уровень устойчивости, что и супер-подстановка AES. Alzette используется для построения малоресурсного 64-битного блочного криптоалгоритма Crax, превосходящего SPECK-64/128 на коротких сообщениях на микроконтроллерах, а также 256-битного блочного шифра Trax.

Минусом подхода, подразумевающего использование ARX-операций являются, как правило, неудовлетворительные криптографические свойства создаваемых конструкций, однако, они позволяют добиться высокого быстродействия и малого потребления ресурсов при программной и аппаратной реализации.

Для противодействия статистическим методам криптоанализа неоднократно осуществлялись попытки применять вместо фиксированных подстановок динамически изменяемые подстановки.

Наиболее успешной попыткой применения динамически изменяемой подстановки можно назвать криптоалгоритм RC4 [26], который считается устаревшим и ненадёжным. Основная проблема стойкости RC4 – применение всего одной динамически изменяемой подстановки и медленное обновление содержимого (за одну итерацию обновляется 2 ячейки из 256) [27]. Проблема предопределена тем, что динамические операции подстановки (в сравнении с фиксированными заменами) требуют на порядки больше вычислительных ресурсов.

Применение псевдо-динамических операций подстановки на базе фиксированных замен потенциально позволяет решить ряд описанных выше проблем, в частности, обеспечить устойчивость к статистическим методам криптоанализа [10].

Исследованию свойств псевдо-динамических подстановок посвящены работы [28–33].

В [28] представлена концепция применения псевдо-динамических таблиц подстановок, позволяющая совместить сильные стороны фиксированных операций подстановки (высокая скорость работы и эффективность использования вычислительных ресурсов) и динамических sbox (нейтрализация статистических методов криптоанализа). Проведенный предварительный анализ линейных и дифференциальных характеристик PD-sbox показал неэффективность их аппроксимации набором линейных функций

и значительное улучшение дифференциальных характеристик при последовательном увеличении количества фиксированных подстановок.

Целью [29] являлась разработка методики определения линейных характеристик псевдо-динамических подстановок для оценки возможности их применения в блочных криптоалгоритмах. Получены выражения для определения линейных свойств псевдо-динамических подстановок PD-sbox для двух случаев: когда значения состояния фиксированы и задаются криптографическим ключом; когда значения состояния динамически изменяются под воздействием энтропии входной информации и результатов предшествующих преобразований. Первичный анализ выражения позволил сделать вывод, что сама структура псевдо-динамической подстановки PD-sbox значительно затрудняет определение её линейных характеристик и, тем самым, препятствует осуществлению линейного криптоанализа.

В [30] цель исследования – определение линейных свойств полноразмерных псевдо-динамических подстановок на основе экстраполяции линейных свойств малоразмерных псевдо-динамических подстановок, сформированных случайным образом. Для упрощения анализа полученных результатов определены усреднённые значения максимумов смещения. Это позволило определить простую закономерность между параметрами PD-sbox и вероятностью получения максимальных значений смещения при случайном формировании PD-sbox. Выявленная закономерность позволила приблизительно экстраполировать линейные свойства малоразмерных псевдо-динамических подстановок на линейные свойства полноразмерных PD-sbox. Проведённая оценка сложности линейного криптоанализа, выраженного в количестве необходимых пар «открытый текст – шифр-текст», показала, что имеется потенциальная возможность синтеза симметричных блочных криптоалгоритмов на основе псевдо-динамических подстановок PD-sbox с экстремально низкими значениями смещения, для которых можно обосновать нижний порог сложности линейного криптоанализа.

Исследованию дифференциальных характеристик псевдо-динамических таблиц подстановок посвящена работа [31]. Определено, что при конкретном значении состояния, порождаемая таблица подстановки с высокой долей вероятности будет невзаимооднозначной и поэтому не может иметь идеальные дифференциальные свойства. Псевдо-динамические операции подстановки, состоящие из взаимооднозначных фиксированных подстановок, обладают идеальными дифференциальными характеристиками при динамическом равновероятном изменении значений состояния. Сформированы дополнительные требования к фиксированным заменам, составляющим псевдо-динамическую операцию подстановки PD-sbox, которые должны быть взаимооднозначными, иметь минимально возможные отклонения дифференциальных характеристик от идеального значения, обладать достаточной нелинейностью с целью разрушения статистических связей между выходными значениями для эффективного изменения значений состояния псевдо-динамических подстановок последующих итераций шифрования. Анализ полученных данных показывает, что поочерёдное добавление в состав PD-sbox фиксированных операций подстановки уменьшает вдвое максимальное значение центрированного коэффициента распространения дифференциалов. В свою очередь, распределение отклонений центрированного коэффициента распространения дифференциалов приближается к гауссовому распределению.

В [32] представлены результаты первоначального анализа псевдо-динамических подстановок, имеющих идеальное распределение дифференциалов, при усреднении всех возможных генерируемых подстановок в статическом режиме работы (при фиксированных значениях состояния). Доказано существование класса псевдо-динамических подстановок PD-sbox, имеющих идеально усредненное распределение дифференциалов в статическом режиме работы. В [33] представлены первые результаты по исследованию нелинейных свойств эквивалентных подстановок, формируемых псевдо-динамическими подстановками,

состоящими из фиксированных операций подстановки размерностью 4 бит. Распределение значений нелинейности для эквивалентных подстановок PD-sbox существенно отличается от распределения значений нелинейности обычных фиксированных подстановок. Примерно 30 полученных псевдо-динамических операций подстановки формируют эквивалентные подстановки с нелинейностью больше нуля. Путём подбора составляющих PD-sbox можно добиться того, что эквивалентные подстановки всегда будут нелинейными.

Развитием структуры псевдо-динамической операции подстановки является применение ARX-функций в их составе. Идея заключалась в том, что объединение слабых, с криптографической точки зрения, конструкций, включающих операции сложения по модулю, циклического сдвига и XOR, позволит получить эквивалентные операции подстановки, обладающие характеристиками, не уступающим случайно сгенерированным операциям замены аналогичной размерности. При этом, полученная структура обладает возможностью параллелизма при её использовании в семействе псевдослучайных функций pCollapser. Одним из основных преимуществ этого подхода является сохранение криптографической устойчивости, при значительном сокращении затрачиваемых ресурсов при программной реализации, а также потенциальное увеличение скорости работы функции, в силу использования более простых операций, в отличие от sbox. В свою очередь, применение подобранных ARX-функций для использования в структуре псевдо-динамических подстановок псевдослучайной функции pCollapser позволяет получить вес дифференциальных и линейных характеристик, превосходящий аналоги, при тех же затратах ресурсов при программной реализации.

Резюме. На текущий момент проблема синтеза операций подстановки как основного нелинейного элемента современных блочных шифров и псевдослучайных функций, удовлетворяющих взаимоисключающим требованиям, является актуальной. Существует ряд способов решения этой проблемы, подразумевающих подбор операций подстановки в соответствии с

требованиями, реализация нелинейного элемента псевдослучайной функции или криптоалгоритма в качестве ARX-функции, применение динамических подстановок в шифрах и синтез псевдо-динамических подстановок, в основе которых могут быть как фиксированные операции замен, так и ARX-конструкции. К операциям подстановки, вне зависимости от их вида, предъявляется около десятка требований [9], напрямую влияющих на криптографическую стойкость псевдослучайных функций, перестановок и криптоалгоритмов. Следовательно, проблема синтеза замен, удовлетворяющих широкому спектру взаимоисключающих параметров является базовой. Существующие подходы к синтезу и применению динамических операций подстановки не позволяют одновременно обеспечить стойкость, минимизацию затрачиваемых ресурсов и скорость программной реализации псевдослучайных функций на их основе, сопоставимую с псевдослучайными функциями на основе фиксированных подстановок или иных фиксированных преобразований.

1.2. Описание структуры псевдо-динамической операции подстановки PD-sbox

Одним из перспективных направлений развития концепции криптографических операций подстановки являются псевдо-динамические операции подстановки PD-sbox, которые объединяют в себе преимущества как фиксированных (высокая скорость работы, эффективное использование вычислительных ресурсов), так и динамических (нейтрализация статистических методов криптоанализа) замен [30].

Структура псевдо-динамической операции подстановки PD-sbox состоит из набора фиксированных замен. Аргумент каждой фиксированной операции подстановки параметризован значением состояния $\mathrm{S_i}$, где i – номер фиксированной подстановки (от 0 до $N-1$).

Текущее значение состояния $\mathrm{S} = \{\mathrm{S_0}, \mathrm{S_1}, \mathrm{S_2}, \dots, \mathrm{S_{N-1}}\}$ задаёт одну операцию подстановки из набора возможных PD-sbox. Подстановка,

полученная при применении конкретного значения состояния, является эквивалентной (сгенерированной). Количество возможных эквивалентных замен определяется набором возможных значений состояния. Данный факт указывает на то, что значение S может динамически изменяться в ходе обработки блоков информации, а вероятностные свойства соответствуют равномерному распределению. Структура псевдо-динамической операции подстановки представлена на рисунке 1.1.

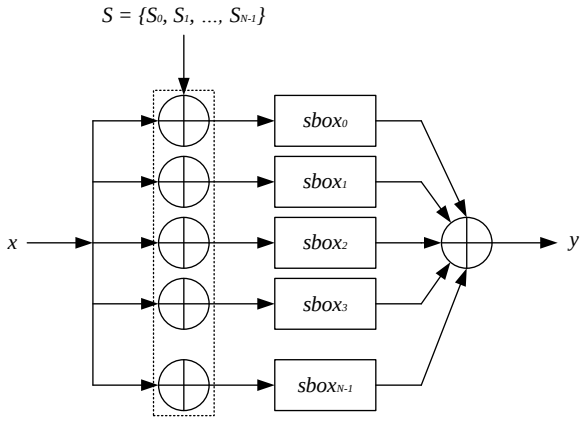


Рисунок 1.1 – Структура PD-sbox

Ниже представлено выражение, описывающее структуру псевдо-динамической операции подстановки PD-sbox:

$$\mathrm{Y} = \oplus_{\mathrm{i}=0}^{\mathrm{N}-1} \mathrm{sbox_i}(\mathrm{X} \oplus \mathrm{S_i}), \quad (1)$$

где $sbox$ – фиксированная операция подстановки; N– количество фиксированных подстановок; X – биты входного сообщения; Y – биты выходного сообщения; S – биты значения состояния псевдо-динамической подстановки; $\oplus$ – операция сложения по модулю 2.

Входное значение каждой фиксированной подстановки задаётся индивидуальным значением состояния S_i, где i – номер фиксированной подстановки (от 0 до $N-1$). Текущее значение состояния $S = \{S_0, S_1, S_2, \dots, S_{N-1}\}$ задаёт одну эквивалентную операцию подстановки из всего множества возможных замен псевдо-динамической подстановки. На рисунке 1.2 представлена псевдо-динамическая операция подстановки в виде набора эквивалентных замен.

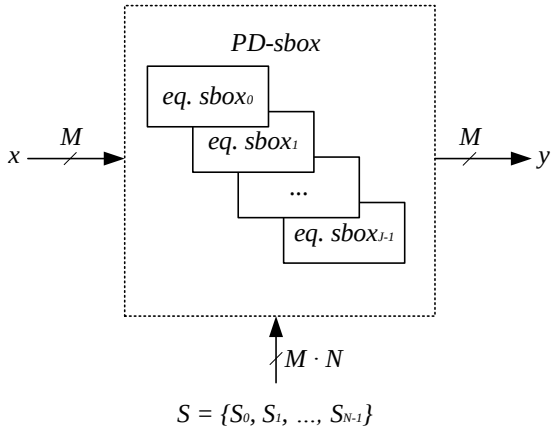


Рисунок 1.2 – Псевдо-динамическая операция подстановки в виде набора эквивалентных замен

Псевдо-динамическая операция подстановки способна функционировать в двух режимах: статическом (ключезависимом) и динамическом (выходное значение зависит не только от ключа, но и от промежуточных состояний).

Статический режим работы подразумевает, что значение внутреннего состояния равно нулю или константе. При динамическом режиме работы наблюдается равновероятное изменение значений внутреннего состояния S и в таком случае дифференциальные усреднённые свойства, а также линейные, близки к идеальным (при усреднении характеристик по всем эквивалентным операциям подстановки). Данная особенность потенциально позволяет нейтрализовать существующие методы дифференциального и линейного криптоанализа [34].

Резюме. Представлено описание структуры псевдо-динамической операции подстановки, а также принцип её работы. PD-sbox обладает преимуществами как фиксированных подстановок (высокая скорость работы, эффективное использование вычислительных ресурсов), так и динамических (нейтрализация статистических методов криптоанализа). Конструкция позволяет обеспечить повышенный параллелизм обработки информации при её использовании в составе псевдослучайных функций, псевдослучайных перестановок и криптоалгоритмов, а также потенциально способна

нейтрализовать существующие методы дифференциального и линейного криптоанализа.

1.3. Линейный криптоанализ PD-sbox на основе фиксированных операций подстановки

Линейным криптоанализом является метод анализа криптоалгоритмов, псевдослучайных функций и псевдослучайных перестановок, заключающийся в поиске наилучших линейных аппроксимаций для отображений, выполняемых шифром или иной функцией [35]. Успех подобной атаки способен значительно снизить сложность поиска ключа, в сравнении с методом полного перебора или «грубой силы».

Для обеспечения достаточного уровня устойчивости к методу линейного криптоанализа, симметричные криптоалгоритмы, а также PRF и PRP строятся по итерационной схеме. Каждый раунд (или итерация) включает в свой состав нелинейные операции и функции перемешивания [33].

Существует ряд базовых методов противодействия линейному криптоанализу:

– увеличение количества раундов шифрования;
– использование более эффективных функций перемешивания;
– увеличение нелинейности фиксированных подстановок.

Основной проблемой нейтрализации линейных атак является особенность всех существующих фиксированных операций подстановки – они не могут обладать идеальными линейными и дифференциальными свойствами, указанный факт в том числе касается использования ARX-функций в качестве операции подстановки. Применение динамических подстановок, по аналогии с алгоритмом RC4, так же не способно решить проблему в силу слабых криптографических свойств [33].

Однако, псевдо-динамические операции подстановки в динамическом режиме работы обладают близкими к идеальным усреднёнными линейными и дифференциальными свойствами, в силу взаимной компенсации значений

смещения при усреднении по всему множеству формируемых эквивалентных подстановок [29].

Исследование методики определения линейных свойств псевдо-динамических подстановок. Основной целью линейного криптоанализа является снижение сложности криптоалгоритма, псевдослучайной функции, псевдослучайной перестановки путём аппроксимации нелинейного элемента линейными статистическими аналогами. В свою очередь, линейные свойства демонстрируют отличие исследуемой функции от набора аффинных преобразований. Метрикой является расстояние Хэмминга – количество отличающихся бит в двух строках одинаковой длины, где строками выступают значения таблиц истинности сравниваемых булевых функций. Сильное соответствие операции подстановки любой из линейных функций позволяет осуществить её замену на линейный статистический аналог, снизив сложность криптоанализа. В случае, если операция подстановки отличается от линейной функции в половине случаев, то её замена линейным статистическим аналогом нецелесообразна в случае равновероятности входных значений [29].

Линейные свойства определяются количеством совпадений нелинейного элемента с набором линейных функций и описываются выражением:

$$NSbox(\alpha,\beta) = \{X|0 \le X < 2^M,\ (\oplus_{i=0}^{M-1}(X[i] * \alpha[i])) = \\ = (\oplus_{j=0}^{N-1}(Sbox(X)[j] * \beta[j]))\}, \quad (2)$$

где $Sbox()$ – выходное значение операции замены; $[j]$ – конкретный бит выходного значения; X – входные значения операции подстановки; $[i]$ – конкретный бит входного значения; 2^M – количество комбинаций на входе подстановки; M – количество бит на входе; N – количество бит на выходе; α – битовая маска для значений на входе; β – битовая маска для значений на выходе; $*$ -- операция побитового логического умножения. Следует отметить, что значения α и β фактически задают вариант линейного статистического аналога [29].

Вероятность замены нелинейного элемента линейной функцией определяется выражением:

$$p(\alpha,\beta) = \frac{NSbox(\alpha,\beta)}{2^M}. \quad (3)$$

Эффективность аппроксимации определяется в виде смещения, демонстрирующего отличие вероятности аппроксимации от идеального значения, равного 0,5.

$$bias(\alpha,\beta) = \left|p(\alpha,\beta) - \frac{1}{2}\right|. \quad (4)$$

При данной методике расчёта линейных свойств идеальным случаем является $bias(\alpha,\beta) = 0$ при всех значениях α и β, кроме нулевых. Очевидно, что наиболее неблагоприятным вариантом значения смещения будет $bias(\alpha,\beta) = 0{,}5$ – полное совпадение нелинейного элемента и линейного статистического аналога [29].

Фиксированная подстановка в блочном криптоалгоритме, над которой осуществляется линейный криптоанализ, может быть описана выражением:

$$Y = Sbox(X \oplus Key). \quad (5)$$

В таком случае достаточно определить линейные свойства непосредственно фиксированной замены sbox, в соответствие с выражением (2), не учитывая вклад значения ключа, так как оно фиксировано и смешивается со значениями на входе сложением по модулю 2. Следовательно, функцию вида:

$$Y = Sbox(X),$$

необходимо аппроксимировать выражением:

$$\oplus_{\mathrm{j}=0}^{N-1}(Y[j] * \beta[j])) = \oplus_{\mathrm{i}=0}^{\mathrm{M}-1}(X[i] * \alpha[i]), \quad (6)$$

где значения α и β задают вариант линейного статистического аналога.

Следовательно, для определения линейных свойств функции (5) требуется:

1. Задать параметры α и β;

2. Последовательно перебрать все значения X и определить для них значения Y;

3. Полученные значения X и Y подставить в выражение (6) и в случае равенства увеличить $NSbox(\alpha, \beta)$ на единицу;

4. Получить матрицу значений $NSbox(\alpha, \beta)$;

5. Вычислить вероятность аппроксимации линейной функцией $p(\alpha, \beta)$ и смещение $bias(\alpha, \beta)$.

6. В матрице $bias(\alpha, \beta)$ осуществить поиск наибольших значений, соответствующих линейным функциям, максимально приближающимся к аппроксимируемой функции [29].

Далее вероятностные значения ключа определяются выражением:

$$\oplus_{\mathrm{i}=0}^{M-1}(Key * \alpha[i]) = (\oplus_{\mathrm{i}=0}^{M-1}(X[i] * \alpha[i])) \oplus (\oplus_{\mathrm{j}=0}^{N-1}(Y[j] * \beta[j])). \quad (7)$$

Анализ линейных свойств псевдо-динамической операции подстановки подразумевает необходимость учёта значения состояния S, задающего порождаемую замену, имеющее значительно большую размерность, чем значения на входе и выходе. Линейный криптоанализ PD-sbox подразумевает два возможных случая:

1. Значения состояний фиксированы. В таком случае псевдо-динамическая операция подстановки может быть представлена в виде большой эквивалентной фиксированной замены, с размерностью входа, соответствующей S.

2. Значения состояний динамически изменяются под воздействием энтропии входных значений и результатов предшествующих преобразований. В этом случае PD-sbox рассматривается как динамически изменяемая замена с размерностями входа и выхода, конкретная подстановка задаётся значением S [29].

Представление псевдо-динамической подстановки в виде большой эквивалентной операции замены. Рассмотрим PD-sbox, состоящий из двух

фиксированных замен. Выражение, определяющее значение на выходе этой функции будет иметь вид:

$$Y = Sbox_0(X \oplus S^0) \oplus Sbox_1(X \oplus S^1). \quad (8)$$

Заменив фиксированные подстановки линейными функциями, получим выражение:

$$\oplus_{j=0}^{N-1}(Y[j] * \beta[j]) = \oplus_{i=0}^{M-1}((X[i] \oplus S^0[i]) * \alpha^0[i]) \oplus$$

$$\oplus \oplus_{i=0}^{M-1}((X[i] \oplus S^1[i]) * \alpha^1[i]),$$

где Y – значение на выходе; $[j]$ – конкретный бит значения на выходе; X – значение на входе; S^0 и S^1 – значения состояния; $[i]$ – конкретный бит значения на входе; M – количество входных бит; N – количество выходных бит; α^0 и α^1 – битовые маски для входных значений; β – битовая маска для значения на выходе; $*$ – операция побитового логического умножения; $\oplus$ – операция XOR.

Далее с определённой вероятностью возможно определить значения бит состояния:

$$(\oplus_{\mathrm{i}=0}^{M-1}(S^0[i] * \alpha^0[i])) \oplus (\oplus_{\mathrm{i}=0}^{M-1}(S^1[i] * \alpha^1[i])) = \quad (9)$$

$$= (\oplus_{\mathrm{j}=0}^{N-1}(Y[j] * \beta[j])) \oplus (\oplus_{\mathrm{i}=0}^{M-1}(X[i] * \alpha^0[i])) \oplus \left(\oplus_{\mathrm{i}=0}^{M-1}\left(X[i] * \alpha^i[i]\right)\right).$$

Алгоритм определения линейных характеристик для статической псевдо-динамической подстановки:

1. Задать значения α^k и β;

2. Последовательно перебрать все значения X и S^i и определить соответствующие значения по формуле (8) на выходе Y;

3. Полученные значения подставить в (1.8). В случае равенства увеличить $NSbox(\alpha, \beta)$ на единицу;

4. Получить матрицу значений $NSbox(\alpha, \beta)$;

5. Вычислить вероятность аппроксимации линейным статистическим аналогом $p(\alpha,\beta)$ и смещение $bias(\alpha,\beta)$. При этом выражение (3) с учётом количества вариантов значения S^i примет вид:

$$p(\alpha,\beta)=\frac{NSbox(\alpha,\beta)}{2^M*\prod_{i=0}^{K-1}2^M}=\frac{NSbox(\alpha,\beta)}{2^{M(1+K)}}, \quad (10)$$

где i – номер фиксированной операции замены, перед которой добавлено значение состояния; M – количество бит в значении состояния S^i; K – количество фиксированных операций замены в псевдо-динамической подстановке.

Выражение, описывающее линейные функции, аппроксимирующие PD-sbox с произвольным количеством фиксированных замен:

$$\oplus_{\mathrm{k=0}}^{K-1}(\oplus_{\mathrm{i=0}}^{M-1}(S^k[i]*\alpha^k[i]))=(\oplus_{\mathrm{j=0}}^{N-1}(Y[j]*\beta[j]))\oplus \quad (11)$$

$$\oplus_{\mathrm{k=0}}^{K-1}(\oplus_{\mathrm{i=0}}^{M-1}(X[i]*\alpha^k[i])),$$

где S^k – значение состояния псевдо-динамической подстановки для k-й фиксированной подстановки; K – количество фиксированных подстановок в PD-sbox; Y – выходное значение; $[j]$ – конкретный бит выходного значения подстановки; X – входное значение; $[i]$ – конкретный бит входного значения фиксированной подстановки; M – количество входных бит; N – количество выходных бит; α^k– битовые маски для входных значений фиксированных подстановок; β – битовая маска для выходного значения; $*$ – операция побитового логического умножения; $\oplus$ – операция XOR [29].

Представление псевдо-динамической подстановки в виде динамически изменяемой операции замены. Режим работы подразумевает динамическое изменение значений состояния S под воздействием энтропии информации на входе, а также результатов предшествующих преобразований псевдослучайной функции, перестановки или криптоалгоритма. Размерность входа и выхода соответствует X и Y, согласно выражению (1). Порождаемая операция подстановки задаётся значением состояния S размерностью $M*K$ бит или $J=2^{M*K}$ комбинаций [29].

В силу равновероятности значений S необходимо усреднять значения для $NSbox(\alpha, \beta)$ по всему множеству эквивалентных подстановок:

$$\overline{NSbox(\alpha, \beta)} = \frac{\sum_{j=0}^{J-1} NSbox_i(\alpha, \beta)}{J}, \quad (12)$$

где $J = 2^{M*K}$ – количество эквивалентных операций подстановки; i – индекс эквивалентной замены; $NSbox_i(\alpha, \beta)$ – матрица совпадений с набором линейных функций для i-ой эквивалентной подстановки; α – битовая маска для значений на входе; β – битовая маска для значений на входе.

После вычисления $NSbox(\alpha, \beta)$ необходимо определить вероятность аппроксимации линейными функциями псевдо-динамической подстановки $p(\alpha, \beta)$ и смещение $bias(\alpha, \beta)$.

Следует отметить, что размерность матрицы $NSbox(\alpha, \beta)$ составляет 2^{M*K} строк и 2^M столбцов. Стандартные фиксированные подстановки, как правило, имеют размерность 2^M строк и 2^M столбцов, значения $M \geq 8$ бит и $K \geq 16$ бит. Это говорит о том, что сложность задачи получения полной матрицы значений $NSbox(\alpha, \beta)$ псевдо-динамической подстановки может превосходить сложность полного перебора ключей криптоалгоритма. Количество комбинаций значений на входе PD-sbox значительно меньше количества значений состояния, что усложняет набор статистики криптоаналитиком и снижает эффективность линейного криптоанализа [29].

Анализ линейных свойств псевдо-динамических подстановок. В ходе диссертационных исследований предложена методика, позволяющая выполнить первичный анализ линейных свойств псевдо-динамических подстановок PD-sbox 6x4x4 и PD-sbox 2x8.

Определение линейных свойств псевдо-динамической подстановки PD-sbox 6x4x4. Для первичного анализа линейных свойств исследована псевдо-динамическая подстановка, содержащая 6 фиксированных операций замены с размерностью входов и выходов 4 бит – PD-sbox 6x4x4:

1. Сформировано 20 случайных PD-sbox. Для этого, случайным образом выбраны 6 фиксированных взаимно однозначных операций подстановки размерностью 4x4 бит;

2. Для каждой из полученной псевдо-динамических операций подстановки случайным образом задано 10000 значений состояний S;

3. Для каждого заданного состояния найдена эквивалентная подстановка и определены её линейные свойства;

4. Подсчитано количество вариантов возникновения нелинейности $N_f = 0$, $N_f = 2$, $N_f = 4$. Наихудшим результатом является $N_f = 0$, это означает, что либо во всех случаях сравнения линейный статистический аналог идентичен эквивалентной подстановке, либо ни в одном из них. Соответственно, наилучший результат $N_f = 4$, когда в половине случаев линейная функция может заменить подстановку, а в половине нет.

5. Для сравнения и первичного набора статистических данных, определены линейные характеристики для 10000 случайных фиксированных взаимно однозначных подстановок аналогичной размерности [33].

Результаты эксперимента на основе представленной методики представлены в таблице 1.1.

Таблица 1.1 – Результаты определения нелинейности N_f

№ PD-sbox	$\boldsymbol{Nf = 0}$	$\boldsymbol{Nf = 2}$	$\boldsymbol{Nf = 4}$
1	149	8535	1316
2	175	8492	1333
3	449	7801	1750
4	158	8443	1398
5	442	7740	1818
6	157	8472	1371
7	145	8495	1360
8	0	8817	1183
9	0	8773	1227
10	0	8763	1237
Random Sbox	**358**	**8764**	**878**

Распределение значений нелинейности для эквивалентных подстановок PD-sbox значительно отличается от распределения значений нелинейности обычных фиксированных операций подстановки.

Примерно 30% полученных псевдо-динамических подстановок формируют эквивалентные замены, для которых нелинейность $N_f > 0$. Для противодействия линейному криптоанализу необходима максимизация этого параметра.

Таким образом, путём подбора составляющих псевдо-динамической операции подстановки можно добиться значительного повышения нелинейности порождаемых эквивалентных подстановок [33].

Определение линейных свойств псевдо-динамической подстановки PD-sbox 2x8. Малый диапазон значений нелинейности эквивалентных подстановок, полученных с помощью PD-sbox, основной которого являются фиксированные операции замены размерностью 4 бит, не позволяет детально оценить особенности распределения нелинейных свойств для эквивалентных подстановок, в том числе сравнить с нелинейностью случайно формируемых операций подстановки [36].

Следующий шаг – используя аналогичную методику, проанализировать линейные свойства эквивалентных операций подстановки, порождаемых псевдо-динамическими подстановками, состоящими из двух 8 битовых фиксированных замен:

1. Случайным образом сформировано 100 псевдо-динамических подстановок PD-sbox-2x8, состоящих из двух фиксированных взаимнооднозначных подстановок размерностью 8x8 бит;

2. Для каждой из полученных PD-sbox-2x8 построена гистограмма распределения значений нелинейности Nf эквивалентных подстановок, получаемых из PD-sbox-2x8 путём подстановки конкретных значений состояния S. Для набора достаточной статистики сформирована 1000 эквивалентных подстановок путём генерации 1000 случайных значений состояний S.

3. Из 100 псевдо-динамических подстановок определены две: с наихудшим распределением и с наилучшим распределение значений нелинейности Nf.

4. Полученные распределения значений нелинейности сравнены с распределением значений нелинейности для 1000 случайных фиксированных невзаимнооднозначных подстановок (сформированных случайным образом) и со значением нелинейности известных фиксированных подстановок из [9].

График распределения значений нелинейности Nf для 1000 небиективных подстановок, сформированных случайным образом, представлен на рисунке 1.3 [36].

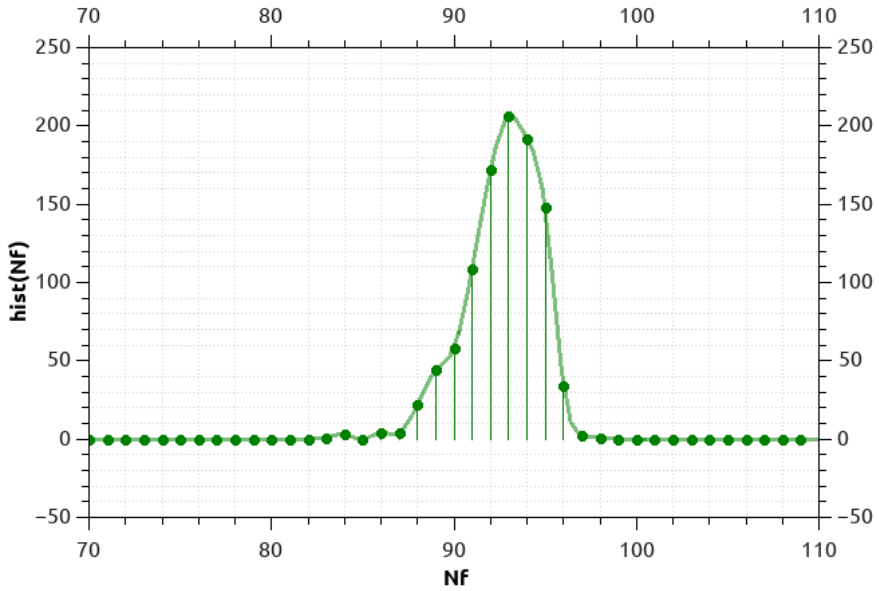


Рисунок 1.3 – Распределение значений нелинейности для 1000 небиективных операций подстановки

Значения нелинейности Nf на полученном графике в основном лежат в диапазоне 88–96, распределение имеет несимметричную форму, что соответствует известным результатам [9]. В свою очередь, на рисунке 1.4 представлена гистограмма распределений значений нелинейности эквивалентных операций подстановки для «наихудшей» (красный) и «наилучшей» (синий) PD-sbox.

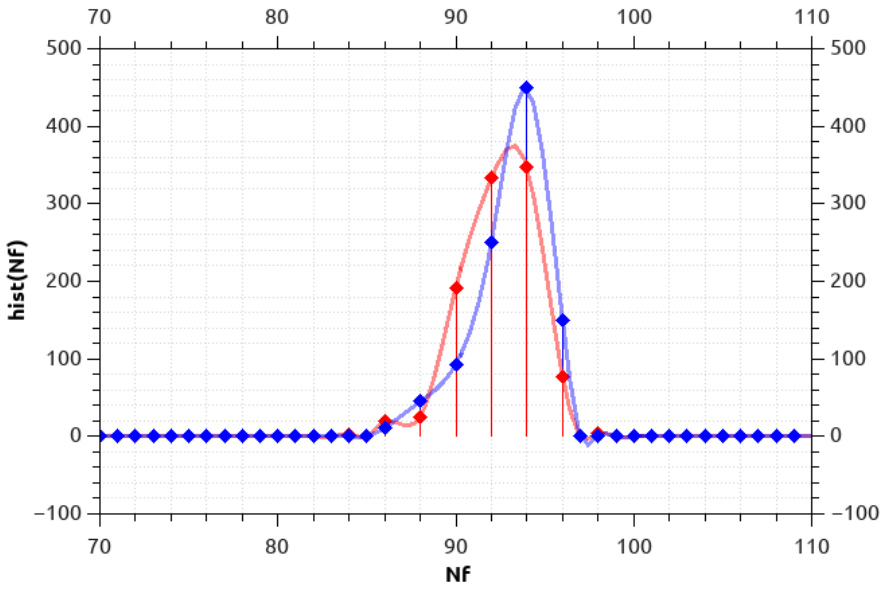


Рисунок 1.4 – Распределения значений нелинейности Nf для эквивалентных подстановок («наихудший» – красный, «наилучший» – синий)

Распределение значений нелинейности Nf для эквивалентных подстановок, формируемых PD-sbox-2x8 соответствует распределению значений нелинейности Nf биективных (взаимнооднозначных подстановок) – т.е. содержат только чётные значения Nf. Это является примечательным, так как сами по себе эквивалентные подстановки являются небиективными (невзаимнооднозначными). Диапазон значений нелинейности Nf для эквивалентных операций подстановки, формируемых PD-sbox-2x8, соответствует диапазону значений нелинейности Nf для случайно формируемых замен. Соответственно, полученные линейные свойства уступают значениям специально созданных подстановок. Так, нелинейность для подстановки из криптоалгоритма AES составляет 112, а значения нелинейности для подстановок из [9] – от 98 до 112. Форма распределения значений нелинейности Nf для эквивалентных подстановок приближается к форме распределения значений нелинейности Nf для случайно формируемых подстановок, но существенно меняется в зависимости от свойств, входящих в состав PD-sbox-2x8 фиксированных операций подстановки.

Таким образом, путём подбора составляющих PD-sbox можно добиться максимизации нелинейности для порождаемых эквивалентных подстановок.

Анализ вычислительно-эффективного метода определения усреднённых линейных свойств псевдо-динамических подстановок. Одной из проблем синтеза псевдо-динамических операций подстановки является отсутствие вычислительно-эффективного метода определения усреднённых линейных свойств для всего множества генерируемых при помощи PD-sbox эквивалентных подстановок, в данном случае под линейными свойствами в первую очередь подразумевается определение максимальных значений смещения $bias(\alpha, \beta)$ от идеального значения, равного 0,5. Для решения этой проблемы предлагается оригинальный метод, состоящий в том, что максимальные значения преобладания рассчитываются только для относительно небольших фиксированных подстановок, входящих в состав PD-sbox, а результирующие максимальные значения преобладания получаются путём итерационного вычисления с использованием логико-вероятностного выражения для операции Исключающего ИЛИ-НЕ (XNOR). Эффектом применения предложенного метода является кардинальное снижение вычислительных операций и, соответственно, возможность определения на типовом персональном компьютере максимальных значений преобладания $bias(\alpha, \beta)$ для 16-элементных псевдо-динамических операций подстановки, состоящих из 8-битовых фиксированных подстановок, что является недостижимым при использовании тривиального метода [37].

Введём следующие обозначения:

K – количество фиксированных подстановок в составе PD-sbox;

M – размерность входа и выхода этих фиксированных подстановок;

N_{rows} – количество строк в таблице $P(\alpha, \beta)$;

$N_{columns}$ – количество столбцов в таблице $P(\alpha, \beta)$;

N_{count} – количество операций подсчёта совпадений одной линейной функции (задаваемой масками α и β) и исследуемой подстановки, соответствует количеству входных комбинаций.

Под вычислительной эффективностью подразумевается количество операций и объём памяти, затрачиваемых при определении линейных свойств операций подстановки [37].

Предлагаемый подход заключается в том, что оцениваются только максимальные значения преобладания $bias(\alpha, \beta)$ каждого из вариантов битовой маски для выходного значения β (т. е., для каждого столбца $NSbox(\alpha, \beta)$ или $P(\alpha, \beta)$), при этом не рассчитываются все варианты битовой маски для входного значения α, имеющего 2^{MK} комбинаций. Вместо этого, вычисляются таблицы $P_i(\alpha^i, \beta)$ для отдельных фиксированных подстановок, а результирующие значения для $P(\alpha, \beta)$ вычисляется с использованием логико-вероятностного выражения, эквивалентному операции Исключающее ИЛИ-НЕ (XNOR) [37].

Рассмотрим пример определения таблицы вероятностей линейной аппроксимации $P(\alpha, \beta)$ для 2-элементной PD-sbox, представленной на рисунке 1.5.

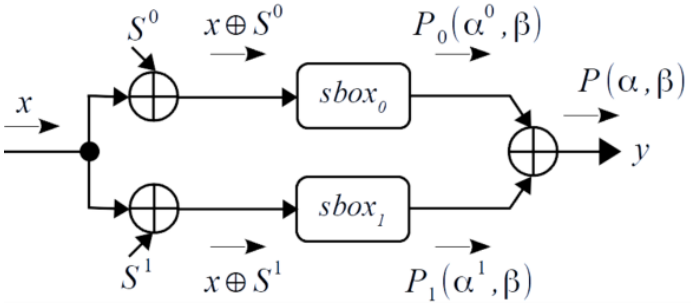


Рисунок 1.5 – Пример 2-элементной PD-sbox

Зададим параметры PD-sbox:

- $N = 3$ – размерность входа, бит;
- $M = 3$ – размерность выхода, бит;
- $K = 2$ – количество фиксированных подстановок;
- $sbox_0(x) = [0, 4, 3, 2, 7, 1, 5, 6]$ – первая подстановка;
- $sbox_1(x) = [5, 0, 4, 3, 2, 1, 6, 7]$ – вторая подстановка.

Определим значения $NSbox(\alpha, \beta)$ и $P(\alpha, \beta)$, используя выражения (2) и (3). Для каждой фиксированной подстановки результаты представлены в таблицах 1.2 и 1.3.

Таблица 1.2 – Значения $NSbox(\alpha, \beta)$ для $sbox_0$ и $sbox_1$

$NS_0(\alpha_0, \beta)$		β							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
α_0	**0**	8	4	4	4	4	4	4	4
	1	4	2	4	6	4	6	4	6
	2	4	4	6	6	4	4	6	2
	3	4	6	2	4	4	6	6	4
	4	4	6	4	6	6	4	2	4
	5	4	4	4	4	6	2	6	6
	6	4	6	6	4	2	4	4	6
	7	4	4	6	2	6	6	4	4

$NS_1(\alpha_1, \beta)$		β							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
α_1	**0**	8	4	4	4	4	4	4	4
	1	4	6	4	2	2	4	2	4
	2	4	4	6	2	6	6	4	4
	3	4	2	2	4	4	6	2	4
	4	4	4	6	6	4	4	2	6
	5	4	2	6	4	2	4	4	2
	6	4	4	4	4	2	6	6	6
	7	4	2	4	2	4	2	4	6

Таблица 1.3 – Значения $P(\alpha, \beta)$ для $sbox_0$ и $sbox_1$

$P_0(\alpha_0, \beta)$

$\alpha_0 \backslash \beta$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.5	0.25	0.5	0.75	0.5	0.75	0.5	0.75
2	0.5	0.5	0.75	0.75	0.5	0.5	0.75	0.25
3	0.5	0.75	0.25	0.5	0.5	0.75	0.75	0.5
4	0.5	0.75	0.5	0.75	0.75	0.5	0.25	0.5
5	0.5	0.5	0.5	0.5	0.75	0.25	0.75	0.75
6	0.5	0.75	0.75	0.5	0.25	0.5	0.5	0.75
7	0.5	0.5	0.75	0.25	0.75	0.75	0.5	0.5

$P_1(\alpha_1, \beta)$

$\alpha_1 \backslash \beta$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.5	0.75	0.5	0.25	0.25	0.5	0.25	0.5
2	0.5	0.5	0.75	0.25	0.75	0.75	0.5	0.5
3	0.5	0.25	0.25	0.5	0.5	0.75	0.25	0.5
4	0.5	0.5	0.75	0.75	0.5	0.5	0.25	0.75
5	0.5	0.25	0.75	0.5	0.25	0.5	0.5	0.25
6	0.5	0.5	0.5	0.5	0.25	0.75	0.75	0.75
7	0.5	0.25	0.5	0.25	0.5	0.25	0.5	0.75

Используя тривиальный метод вычислим большую эквивалентную подстановку, соответствующую двум параллельно включенным фиксированным подстановкам. Для этого переберём все возможные входные комбинации $x \oplus S^0 \,||\, x \oplus S^1$ и вычислим соответствующие выходные

значения y. Размерность входа составит $N = 3 \cdot 2$ или $N = 6$ бит, а размерность выхода будет $M = 3$ бита [37].

В нашем случае большая эквивалентная подстановка будет иметь вид:

$$bigSbox(x) = [5, 1, 6, 7, 2, 4, 0, 5, 1, 6, 7, 2, 4, 0, 3, 0, 4, 3, 2, 7,$$

$$1, 5, 6, 4, 0, 7, 6, 3, 5, 1, 2, 3, 7, 0, 1, 4, 2, 6, 5, 2, 6, 1,$$

$$0, 5, 3, 7, 4, 1, 5, 2, 3, 6, 0, 4, 7, 6, 2, 5, 4, 1, 7, 3, 0, 7, 3, 4, 5, 0, 6, 2, 1].$$

Определим значения $NSbox(\alpha, \beta)$ и $P(\alpha, \beta)$ для большой эквивалентной подстановки $bigSbox$. Результаты представлены в таблицах 1.4 и 1.5.

Таблица 1.4 – Значения $NSbox(\alpha, \beta)$ для $bigSbox$

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7
0	64	32	32	32	32	32	32	32
1	32	32	32	32	32	32	32	32
…	…	…	…	…	…	…	…	…
9	32	24	32	24	32	32	32	32
10	32	32	32	24	32	32	24	32
11	32	40	32	32	32	32	24	32
12	32	40	32	24	24	32	40	32
…	…	…	…	…	…	…	…	…
63	32	32	32	40	32	24	32	32

Таблица 1.5 – Значения $P(\alpha, \beta)$ для $bigSbox$

$\alpha \backslash \beta$	0	1	2	3	4	5	6	7
0	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
…	…	…	…	…	…	…	…	…
9	0.5	0.375	0.5	0.375	0.5	0.5	0.5	0.5
10	0.5	0.5	0.5	0.375	0.5	0.5	0.375	0.5
11	0.5	0.625	0.5	0.5	0.5	0.5	0.375	0.5
12	0.5	0.625	0.5	0.375	0.375	0.5	0.625	0.5
…	…	…	…	…	…	…	…	…
63	0.5	0.5	0.5	0.625	0.5	0.375	0.5	0.5

Обратим внимание на следующие значения $P(\alpha,\beta)$ для $bigSbox$:

$$\alpha = 9; \beta = 4; P(9,4) = 0.5,$$

$$\alpha = 9; \beta = 1; P(9,1) = 0.375,$$

$$\alpha = 11; \beta = 1; P(11,1) = 0.625.$$

Согласно указанному принципу вычисления $bigSbox$ соответствие между значениями масок $bigSbox$, $sbox_0$ и $sbox_1$ будет иметь следующий вид:

$$\alpha = \{\alpha^0 || \alpha^1\} = \alpha^0 \cdot 2^M + \alpha^1, \quad (13)$$

где $||$ – операция конкатенации двухбитовых слов.

Таким образом, мы имеем следующее соответствие между масками:

$$\alpha = 9 = 1 \cdot 8 + 1 \rightarrow \alpha^0 = 1, \alpha^1 = 1,$$

$$\alpha = 9 = 1 \cdot 8 + 1 \rightarrow \alpha^0 = 1, \alpha^1 = 1,$$

$$\alpha = 11 = 1 \cdot 8 + 3 \rightarrow \alpha^0 = 1, \alpha^1 = 3.$$

С учётом этого, указанные выше значения $P(\alpha,\beta)$ для $bigSbox$ зависят от следующих значений $P(\alpha,\beta)$ фиксированных подстановок $sbox_0$ и $sbox_1$:

$$P(9,4) = 0.5 : P_0(\alpha^0 = 1, \beta = 4) =$$

$$= 0.5 \text{ и } P_1(\alpha^1 = 1, \beta = 4) = 0.25,$$

$$P(9,1) = 0.375 : P_0(\alpha^0 = 1, \beta = 1) =$$

$$= 0.25 \text{ и } P_1(\alpha^1 = 1, \beta = 1) = 0.75,$$

$$P(11,1) = 0.625 : P_0(\alpha^0 = 1, \beta = 1) =$$

$$= 0.25 \text{ и } P_1(\alpha^1 = 3, \beta = 1) = 0.25.$$

Обозначим функцию, которая связывает $P(\alpha,\beta)$ с $P_0(\alpha^0,\beta)$ и $P_1(\alpha^1,\beta)$ как $F()$:

$$P(\alpha,\beta) = F(P_0(\alpha^0,\beta), P_1(\alpha^1,\beta)). \quad (14)$$

Тогда приведённые выше варианты можно записать в следующем виде:

$$F(0{,}5; 0{,}25) = 0{,}5,$$

$$F(0{,}25; 0{,}75) = 0{,}375,$$

$$F(0{,}25; 0{,}25) = 0{,}625.$$

Выражение для первого случая сразу наводит на мысль, что наблюдается зависимость, аналогичная операции XOR (Исключающее ИЛИ). Как известно [38], логико-вероятностное выражение для XOR является «терминатором» – если на любом из входов будет равновероятное значение ($x = 0{,}5$), то на выходе также будет равновероятное значение: $Pxor(0{,}5; any) = 0{,}5$.

Выражения для второго и третьего случая позволяют уточнить вид зависимости. Найденное выражение, описывающее зависимость между $P(\alpha, \beta)$, $P_0(\alpha^0, \beta)$ и $P_1(\alpha^1, \beta)$, имеет следующий вид:

$$p(\alpha, \beta) = F(p_0, p_1) = 1 - ((1 - p_0) \cdot p_1 + p_0 \cdot (1 - p_1)), \quad (15)$$

где $p_0 = P_0(\alpha^0, \beta)$; $p_1 = P_1(\alpha^1, \beta)$.

Легко проверить, что данное выражение соответствует операции Исключающее ИЛИ-НЕ (XNOR) – путём подстановки значений «0» и «1» в выражение для $F(p_0, p_1)$ в соответствии с таблицей истинности XNOR.

То, что операции XOR на выходе PD-sbox соответствует логико-вероятностное выражение само по себе не вызывает вопросы. Теория логико-вероятностных выражений развивается много лет и находит применение, в том числе, для расчёта надёжности сложных систем [38].

Очень интересным фактом выступает то, что здесь логико-вероятностное выражение описывает связь между вероятностями аппроксимации подстановок линейными функциями, причём эта зависимость имеет инверсный характер – на выходе PD-sbox расположена операция XOR, а выражение для $P(\alpha, \beta)$ соответствует логико-вероятностной форме операции XNOR, представленной на рисунке 1.6 [37].

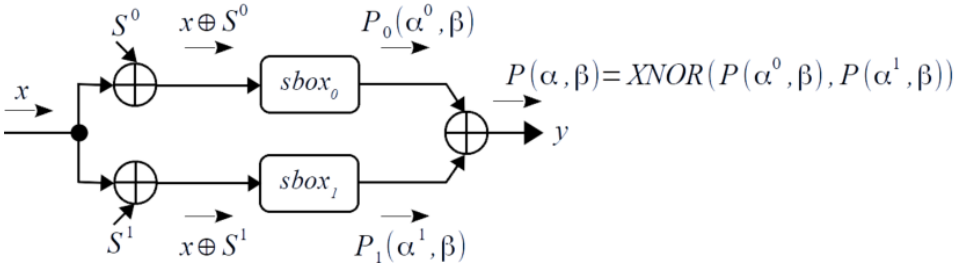


Рисунок 1.6 – Пояснения формирования $P(\alpha, \beta)$

Таким образом, получен первый важный вывод: для вычисления таблицы вероятностей $P(\alpha, \beta)$ для 2-элементной PD-sbox достаточно только таблиц вероятностей $P_0(\alpha^0, \beta)$ и $P_1(\alpha^1, \beta)$ соответствующих фиксированных подстановок $sbox_0$ и $sbox_1$ (входящих в состав PD-sbox) и вычисления по формуле (26) результирующих значений таблицы вероятностей $P(\alpha, \beta)$.

Однако, с точки зрения вычислительной эффективности данный вывод пока не даёт преимуществ, так как для расчёта результирующей таблицы вероятностей $P(\alpha, \beta)$ потребуется проход 2^{2M} значений $\alpha = \alpha^0 || \alpha^1$.

Рассмотрим пример определения таблицы вероятностей линейной аппроксимации $P(\alpha, \beta)$ для 3-элементной PD-sbox. В соответствии с правилами булевой алгебры, мы можем представить операцию XOR от 3 переменных в виде двух последовательных операций XOR от 2 переменных. Данный вариант PD-sbox представлен на рисунке 1.7 [37].

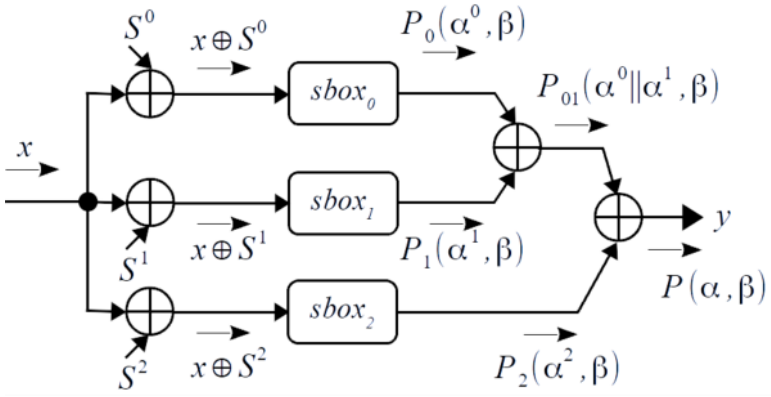


Рисунок 1.7 – Пример 3-элементной PD-sbox

Проведённые исследования показывают, что для этого случая также походит выражение (15), только в качестве p_0 подставляется результат

вычисления $P_{01}(\alpha^0 \parallel \alpha^1, \beta) = F(P_{01}(\alpha^0, \beta), P_1(\alpha^1, \beta))$ для двух фиксированных подстановок:

$$P(\alpha, \beta) = F(p_0, p_1) = 1 - ((1 - p_0) \cdot p_1 + p_0 \cdot (1 - p_1)), \quad (16)$$

где $p_0 = P_{01}(\alpha^0 \parallel \alpha^1, \beta)$; $p_1 = P_2(\alpha^2, \beta)$.

Очевидно, что таким итерационным способом возможно определить значения таблицы вероятностей $P(\alpha, \beta)$ для K-элементных PD-sbox. Таким образом, получен второй важный вывод: для вычисления таблицы вероятностей $P(\alpha, \beta)$ для K-элементных PD-sbox достаточно только таблиц вероятностей $P_0(\alpha^0, \beta) \ldots P_{K-1}(\alpha^{K-1}, \beta)$ соответствующих фиксированных подстановок $sbox_0 \ldots sbox_{K-1}$ (входящих в состав PD-sbox) и итерационного попарного вычисления по формуле (16) значений таблицы вероятностей $P(\alpha, \beta)$ [37].

Итоговое итерационное выражение, позволяющее определить результирующее значения вероятностей $P(\alpha, \beta)$ для K-элементных PD-sbox, можно записать в следующем виде:

$$P\left(\alpha^0 \parallel \alpha^i, \beta\right)|_{imax=K-1} = \quad (17)$$

$$= 1 - \left(1 - P(\alpha^0 \parallel \alpha^{i-1}, \beta)\right) \cdot P\left(\alpha^i, \beta\right) +$$

$$+ P(\alpha^0 \parallel \alpha^{i-1}, \beta) \cdot \left(1 - P\left(\alpha^i, \beta\right)\right).$$

Как и в предыдущем случае, с точки зрения вычислительной эффективности данный вывод пока не даёт преимуществ, так как для расчёта результирующей таблицы вероятностей $P(\alpha, \beta)$ потребуется проход 2^{KM} значений маски $\alpha = \alpha^0 \parallel \alpha^1 \parallel \ldots \parallel \alpha^{K-1}$.

С точки зрения стойкости к линейному криптоанализу в большинстве случаев важно только определение максимальных отклонений значений преобладания $bias(\alpha, \beta)$ от идеального значения $0{,}5$.

Значения преобладания $bias(\alpha, \beta)$ показывают отклонение вероятности аппроксимации подстановки линейными функциями $P(\alpha, \beta)$ от равновероятного значения $0{,}5$. Если проанализировать выражение (15), то

максимальное значение на выходе $F(p_0, p_1)$ будет в случае, если на входах будут значения p_0 и p_1, максимально отличающиеся от значения 0,5. Иными словами, максимальное значение $bias(\alpha, \beta)$ задаётся максимальными значениями исходных фиксированных подстановок $bias(\alpha^0, \beta)$ и $bias(\alpha^1, \beta)$.

Вернёмся к случаю с 2-элементными PD-sbox. Так как для всех вариантов выходной маски β нужны максимальные значения $bias(\alpha, \beta)$, то для поиска максимальных значений $bias(\alpha, \beta)$ вместо полных таблиц $bias(\alpha^0, \beta)$ и $bias(\alpha^1, \beta)$ достаточно найти по одной строке с максимальными значениями из этих таблиц [37]. Например:

$$row_{maxbias}(\alpha^0, \beta) = \quad (18)$$

$$= \{bias_{max}(\alpha^0, 0); bias_{max}(\alpha^0, 1); bias_{max}(\alpha^0, 2); \ldots; bias_{max}(\alpha^0, 2^{M-1})\},$$

$$row_{maxbias}(\alpha^1, \beta) = \quad (19)$$

$$= \{bias_{max}(\alpha, 0); bias_{max}(\alpha^1, 1); bias_{max}(\alpha^1, 2); \ldots; bias_{max}(\alpha^1, 2^{M-1})\}.$$

Для 2-элементной PD-sbox строки будут иметь вид, представленный в таблице 1.6:

Таблица 1.6 – Строки 2-элементной PD-sbox

$P_{max}(\alpha^0, \beta)$

$\boldsymbol{\alpha_0 \backslash \beta}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
max	0.5	0.25	0.75	0.75	0.75	0.75	0.75	0.75

$P_{max}(\alpha^1, \beta)$

$\boldsymbol{\alpha_1 \backslash \beta}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
max	0.5	0.75	0.75	0.25	0.75	0.75	0.25	0.75

$bias(\alpha^0, \beta) = |P_0(\alpha^0, \beta) - 0{,}5|$

$\boldsymbol{\alpha_0 \backslash \beta}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
max	0	0.25	0.25	0.25	0.25	0.25	0.25	0.25

$bias(\alpha^1, \beta) = |P_1(\alpha^1, \beta) - 0{,}5|$

$\boldsymbol{\alpha_1 \backslash \beta}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
max	0	0.25	0.25	0.25	0.25	0.25	0.25	0.25

При попарной подстановке значений $P_{max}(\alpha^0, \beta)$ и $P_{max}(\alpha^1, \beta)$ в выражение XNOR (15) получена строка с максимальными значениями $P_{max}(\alpha^0 \parallel \alpha^1, \beta)$:

Или, если перевести в значения $bias_{max}(\alpha^0 \parallel \alpha^1, \beta)$, значения представлены в таблице 1.7.

Таблица 1.7 – Значения $bias_{max}(\alpha^0 \parallel \alpha^1, \beta)$

$\alpha^0 \parallel \alpha^1 \backslash \boldsymbol{\beta}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
max	0,5	0,375	0,625	0,375	0,625	0,625	0,375	0,625

Полученные максимальные значения совпадают с максимальными значениями при расчёте полных таблиц $P(\alpha, \beta)$, $P_0(\alpha^0, \beta)$ и $P_1(\alpha^1, \beta)$.

Используя выражение (17), приведённый пример можно расширить на вычисление максимальных значений $P_{max}(\alpha, \beta)$ и $bias_{max}(\alpha, \beta)$ для K-элементных подстановок PD-sbox:

$$P_{max}(\alpha^0 \parallel \alpha^i, \beta)|_{imax=K-1} = 1 - ((1 - P_{max}(\alpha^0 \parallel \alpha^{i-1}, \beta)) \cdot P_{max}(\alpha^i, \beta) + \quad (20)$$
$$+P_{max}(\alpha^0 \parallel \alpha^{i-1}, \beta) \cdot (1 - P_{max}(\alpha^i, \beta))).$$

Для предложенного метода вычислительные затраты будут складываться из следующих составляющих:

1. Расчёт таблиц $P_0(\alpha^0, \beta) \ldots P_{K-1}(\alpha^{K-1}, \beta)$ и преобладания $bias_0(\alpha^0, \beta) \ldots bias_{K-1}(\alpha^{K-1}, \beta)$ для фиксированных подстановок, входящих в состав PD-sbox. Для одной фиксированной подстановки потребуется следующее число проходов:

$$N_{Sbox} = N_{rows} \cdot N_{columns} \cdot N_{count} = 2^M \cdot 2^M \cdot 2^M = 2^{3 \cdot M}. \quad (21)$$

2. Поиск максимальных значений по столбцам в таблице $bias(\alpha, \beta)$ и составление строки максимальных значений. Для одной фиксированной подстановки всего требуется $N_{find\_max} = N_{rows} \cdot N_{columns} = 2^M \cdot 2^M = 2^{2 \cdot M}$ операций просмотра и определения максимальных значений.

3. Итоговый расчёт вероятности $P(\alpha, \beta)$ и преобладания $bias(\alpha, \beta)$ используя итерационное выражение (20). Для K-элементной PD-sbox всего потребуется $N_{iter} = K - 1$ итераций вычисления функции Исключающее ИЛИ-НЕ $F(\alpha, \beta)$.

Например, для PD-sbox с $K = 8$ и $M = 4$ мы получим $K \cdot N_{Sbox} = K \cdot 2^{3 \cdot M} = 8 \cdot 2^{12}$ проходов для вычисления таблиц $NSbox(\alpha, \beta)$ всех фиксированных подстановок. Такое же количество уйдёт на вычисление значений вероятностей и преобладания.

Кроме этого, потребуется $K \cdot N_{find\_max} = K \cdot 2^{2 \cdot M} = 8 \cdot 2^8$ операций просмотра и определения максимальных значений для всех фиксированных подстановок и $N_{iter} = 7$ итераций вычисления функции Исключающее ИЛИ-НЕ $F(\alpha, \beta)$.

В таблице 1.8 приведено сравнение тривиального и предложенного методов по двум наиболее ресурсоёмким показателям.

Таблица 1.8 – Эффективность предложенного метода поиска $bias_{max}(\alpha, \beta)$

	Тривиальный метод		**Предложенный метод**	
PD-sbox:	$N_{bigSbox}$	$sizeNsbox(\alpha, \beta)$	$K \cdot N_{Sbox}$	$sizeNsbox(\alpha, \beta)$
$K = 2$ и $M = 4$	2^{20}	$2^8 \times 2^4$	2^{13}	$2 \times 2^4 \times 2^4$
$K = 4$ и $M = 4$	2^{36}	$2^{16} \times 2^4$	2^{14}	$4 \times 2^4 \times 2^4$
$K = 8$ и $M = 4$	2^{68}	$2^{32} \times 2^4$	2^{15}	$8 \times 2^4 \times 2^4$
$K = 16$ и $M = 4$	2^{132}	$2^{64} \times 2^4$	2^{16}	$16 \times 2^4 \times 2^4$
$K = 2$ и $M = 8$	2^{40}	$2^{16} \times 2^8$	2^{25}	$2 \times 2^8 \times 2^8$
$K = 4$ и $M = 8$	2^{72}	$2^{32} \times 2^8$	2^{26}	$4 \times 2^8 \times 2^8$
$K = 8$ и $M = 8$	2^{136}	$2^{64} \times 2^8$	2^{27}	$8 \times 2^8 \times 2^8$
$K = 16$ и $M = 8$	2^{264}	$2^{128} \times 2^8$	2^{28}	$16 \times 2^8 \times 2^8$

Как видно, данная задача является решаемой при использовании типовых персональных компьютеров.

Таким образом, представлен вычислительно эффективный метод определения усреднённых линейных свойств псевдо-динамических подстановок, заключающийся в поиске максимальных значений преобладания (смещения) вероятности линейной аппроксимации $bias(\alpha, \beta)$ для K-элементных PD-sbox, который состоит из следующих этапов:

1. Расчёт таблиц значений вероятности $P_0(\alpha^0, \beta) \ldots P_{K-1}(\alpha^{K-1}, \beta)$ и преобладания $bias_0(\alpha^0, \beta) \ldots bias_{K-1}(\alpha^{K-1}, \beta)$ для фиксированных подстановок, входящих в состав PD-sbox.

2. Для каждой из полученных таблиц $bias_0(\alpha^0, \beta) \ldots bias_{K-1}(\alpha^{K-1}, \beta)$ формируется строка максимальных значений $bias_{maxi}(\alpha^i, \beta)$, содержащая максимальные значения преобладания для всех комбинаций маски b.

3. Для каждой из полученных таблиц $P_0(\alpha^0, \beta) \ldots P_{K-1}(\alpha^{K-1}, \beta)$ формируется аналогичная строка значений $P_{maxi}(\alpha^i, \beta)$, с соответствующими п.2 значениями вероятностей.

4. Расчёт промежуточных и итоговых максимальных значений $P_{max}(\alpha, \beta)$ и $bias_{max}(\alpha, \beta)$ используя итерационное выражение (31).

Предложенный метод, в противовес известным подходам, позволяет определять максимальные значения $bias_{max}(\alpha, \beta)$ используя приемлемые вычислительные ресурсы [37].

Резюме. Представлена методика определения линейных свойств псевдо-динамических операций подстановки, основанных на фиксированных заменах. Обозначена разница в подходе к линейному криптоанализу PD-sbox, в зависимости от режима её работы. Стоит отметить, что порождаемые PD-sbox эквивалентные подстановки не независимы, следовательно, рассматривать их статистические свойства по отдельности не корректно. Проведён первичный анализ линейных свойств эквивалентных подстановок для PD-sbox 6x4x4 и 2x8. Результат показал, что при случайном выборе фиксированных подстановок, на базе которых строится PD-sbox, свойства эквивалентных подстановок не уступают случайно сгенерированным заменам той же размерности. Сделаны выводы, что при специальном подборе фиксированных замен потенциально возможно максимизировать нелинейность эквивалентных операций подстановки PD-sbox. Представлен вычислительно-эффективный метод поиска усреднённых линейных свойств псевдо-динамических подстановок, позволяющий определять максимальные значения смещения, используя приемлемые вычислительные ресурсы, в противовес известным подходам.

1.4. Дифференциальный криптоанализ PD-sbox на основе фиксированных операций подстановки

Дифференциальным криптоанализом является анализ криптоалгоритмов, псевдослучайных функций и псевдослучайных

перестановок, заключающийся в поиске наиболее вероятных разностей, получаемых на выходе шифра или иной функции при заданной входной разности [35].

Дифференциальный криптоанализ, как и линейный, является статистическим методом криптоанализа, основной целью является поиск взаимосвязей между разностями сообщений на входе ΔX и на выходе ΔY [39]. Разность между двумя двоичными строками Z' и Z'' представляют в виде:

$$\Delta Z = Z' \oplus Z''. \quad (22)$$

Преимуществом дифференциальной атаки является то, что разность без изменений проходит через операцию сложения по модулю 2, применяемой для смешивания значений ключа и обрабатываемой информации. Эта особенность позволяет существенно упростить анализ распространения дифференциалов в блочных криптоалгоритмах [31].

Блочный шифр, обладающий идеальными дифференциальными свойствами, должен обеспечивать вероятность появления на выходе разности ΔY при заданной входной разности ΔX, равную $(1/2)^M$, где M – количество бит в сравниваемых битовых строках.

При дифференциальном криптоанализе определяют значение $N_D(\Delta X, \Delta Y)$, демонстрирующее количество пар $\{\Delta X; \Delta Y\}$. Для идеальных фиксированных операций подстановки

$$N_D(\Delta X, \Delta Y) = 1. \quad (23)$$

Стоит отметить, что подобные фиксированные операции замены математически невозможны [39].

На основе N_D рассчитывается коэффициент распространения дифференциалов:

$$R_P(\Delta X, \Delta Y) = N_D(\Delta X, \Delta Y)/2^M. \quad (24)$$

Основной задачей при проведении дифференциального криптоанализа является поиск дифференциалов с максимальным отклонением от идеального

значения. В свою очередь, разработчики криптоалгоритмов, псевдослучайных функций или псевдослучайных перестановок используют операции подстановки с минимальным отклонением дифференциалов от идеального (равновероятного) значения [31].

При осуществлении дифференциального криптоанализа псевдо-динамических операций подстановки задача усложняется, так как первое значение разности ΔX пройдёт через одну эквивалентную замену, а второе значение через другую.

Существуют два крайних случая:

1. Значения состояния S фиксированы и задаются ключом. Значения разности ΔX проходят через одну эквивалентную таблицу подстановки. Этот вариант аналогичен случаю с обычной фиксированной операцией подстановки. Однако, поскольку значение состояния S криптоаналитику не известно, то и дифференциальные свойства необходимо усреднять по всем возможным равновероятным значениям параметра S (исходя из равновероятности значений ключа шифрования) [31]:

$$\overline{N_D}(\Delta X, \Delta Y) = \frac{\sum_{i=0}^{N-1} N_{Di}(\Delta X, \Delta Y_i)}{N}, \quad (25)$$

$$\Delta X = X' \oplus X'', X' \in \{0 \dots 2^M\}, X'' \in \{0 \dots 2^M\}, \quad (26)$$

$$\Delta Y_i = Sbox_i(X') \oplus Sbox_i(X''), \quad (27)$$

где $N_{Di}(\Delta X, \Delta Y_i)$ – значение $N_D(\Delta X, \Delta Y)$ для i-й эквивалентной замены; N – количество порождаемых операций подстановки.

2. Значения состояния динамически изменяются и зависят от ключа и текущих значений операций преобразования данных. Пусть известно количество порождаемых эквивалентных замен N. В этом случае дифференциальные свойства определяются с учётом прохождения частей разности ΔX через разные эквивалентные (порождаемые) операции подстановки:

$$N_D(\Delta X, \Delta Y) = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} N_{Dij}(\Delta X, \Delta Y_{ij})}{N^2}, \quad (28)$$

$$\Delta X = X' \oplus X'', \quad (29)$$

$$\Delta Y_{ij} = Sbox_i(X') \oplus Sbox_j(X''), \quad (30)$$

где $N_{Dij}(\Delta X, \Delta Y_{ij})$ – значение $N_D(\Delta X, \Delta Y)$ в случае, если первая компонента ΔX проходит через i-ю эквивалентную подстановку $Sbox_i()$, а вторая компонента ΔX проходит через j-ю эквивалентную замену $Sbox_j()$.

Эксперимент показал, что псевдо-динамические операции подстановки, которые состоят из взаимнооднозначных фиксированных замен, обладают идеальными дифференциальными характеристиками при динамическом равновероятном изменении значений состояния S. Эффект проявляется при любом количестве взаимооднозначных фиксированных операций подстановки, входящих в состав PD-sbox [31].

Пусть значения состояния фиксированы и задаются равновероятным ключом. Случай характерен для первых итераций преобразования до момента набора достаточной энтропии изменения параметров S, задающих последующие эквивалентные операции подстановки, порождаемые PD-sbox [31].

Элементами PD-sbox выбраны 4 битные фиксированные операции подстановки, выбор обусловлен частым применениям в легковесных криптоалгоритмах и неудовлетворительными статистическими свойствами, в сравнении с большими размерностями. Оценим влияние на дифференциальные свойства 4-х разрядной псевдо-динамической операции подстановки PD-sbox количества фиксированных замен, входящих в её состав. Случайным образом сгенерируем фиксированные операции подстановки $sbox_0 \ldots sbox_4$, значения которых приведены в таблице 1.9.

Таблица 1.9 – Фиксированные замены для исследуемых вариантов PD-sbox

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\boldsymbol{sbox_0(X)}$	5	B	E	F	6	A	0	8	C	D	1	4	2	9	7	3
$\boldsymbol{sbox_1(X)}$	7	8	6	0	B	E	A	C	5	4	1	9	D	F	3	2
$\boldsymbol{sbox_2(X)}$	E	5	4	F	8	0	6	D	A	7	3	B	2	C	1	9
$\boldsymbol{sbox_3(X)}$	4	5	1	B	E	3	6	2	7	0	A	C	9	D	F	8
$\boldsymbol{sbox_4(X)}$	2	6	5	1	C	B	9	3	8	0	D	F	A	7	E	4

Подобные операции подстановки выбраны с минимально возможными дифференциальными свойствами для демонстрации возможностей PD-sbox по улучшению дифференциальных характеристик. Расчёт дифференциальных характеристик для большего количества фиксированных операций подстановки ограничен наличием имеющихся вычислительных ресурсов [31].

Определены дифференциальные характеристики PD-sbox для случаев с 1-й, 2-я, 3-я и 4-я фиксированными операциями подстановки. Рисунки 1.8 и 1.9 демонстрируют варианты PD-sbox для случаев $K = 1$ и $K = 3$. Для каждого варианта определены усреднённые значения числа дифференциалов $N_D(\Delta X, \Delta Y)$, согласно (25). В соответствии с (24) найдены усреднённые значения коэффициента распространения дифференциалов $R_P(\Delta X, \Delta Y)$. Значение коэффициента распространения $R_P(\Delta X, \Delta Y)$ отцентрированы относительно идеального значения $R_P = 1 / 2^4 = 0{,}0625$ [31].

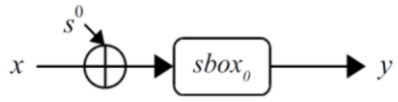


Рисунок 1.8 – PD-sbox на базе одной фиксированной замены

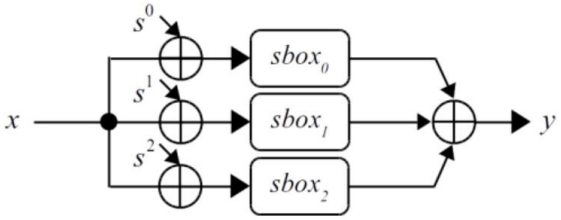


Рисунок 1.9 – PD-sbox, состоящая из трёх фиксированных операций подстановки

Таблица 1.10 демонстрирует максимальные центрированные отклонения, рисунок 1.10 представляет диаграмму центрированных

отклонений коэффициента распространения дифференциалов $R_P(\Delta X, \Delta Y)$ при различном числе фиксированных операций подстановки. Исключены значения для случая $\Delta X = 0$, не представляющего интерес для дифференциального криптоанализа [31].

Таблица 1.10 – Максимальные центрированные отклонения коэффициента распространения дифференциалов

Количество таблиц подстановок, K	**1**	**2**	**3**	**4**	**5**	**AES sbox**
Максимальное отклонение $R_P(\Delta X, \Delta Y)$ от идеального значения	0,313	0,156	0,039	0,016	0,004	0,012

Рисунок 1.10 демонстрирует, что добавление в PD-sbox дополнительных фиксированных замен в два раза снижает максимальное значение центрированного коэффициента распространения дифференциалов $R_P(\Delta X, \Delta Y)$, а также распределение отклонений центрированного коэффициента распространения дифференциалов приближается к гауссовому распределению [22].

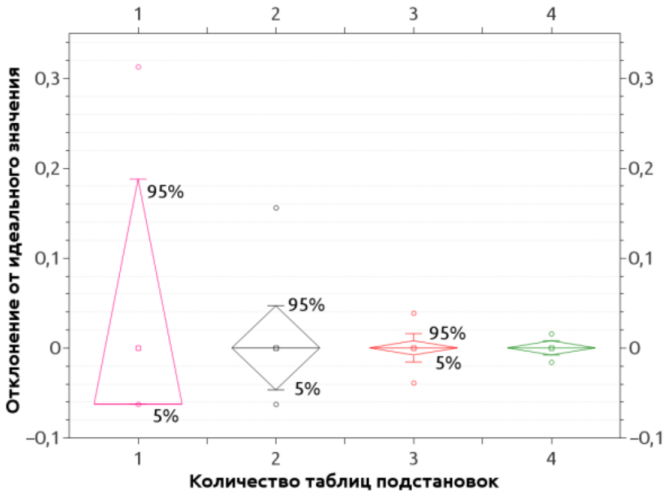


Рисунок 1.10 – Диаграмма распределения отклонений коэффициента распространения дифференциалов относительно идеального значения для различного количества фиксированных операций подстановки

На рисунке 1.11 представлена диаграмма центрированных отклонений коэффициента распространения дифференциалов для фиксированной

операции замены из криптоалгоритма AES и PD-sbox, состоящей из 4-х и 5-и фиксированных подстановок. Стоит отметить, что операция подстановки криптоалгоритма AES имеет размерность 8 бит и специально подобрана и обладает лучшими статистическими свойствами, в сравнении с рассматриваемыми. Для фиксированной замены AES идеальное значение коэффициента распространения $R_P(\Delta X, \Delta Y) = 1 / 2^8 = 0{,}004$, а максимальное значение отклонения $max(\, R_P(\Delta X, \Delta Y) - 0{,}004\,) = 0{,}012$ [31].

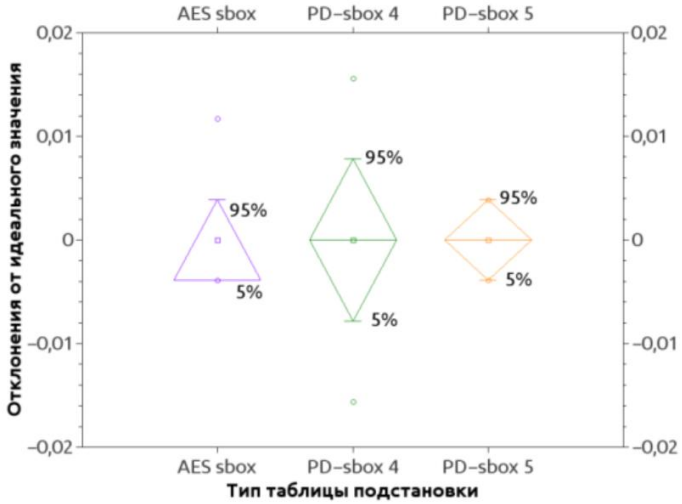


Рисунок 1.11 – Диаграмма распределения отклонений коэффициента распространения дифференциалов относительно идеального значения

Рисунок 1.11 демонстрирует, что в случае фиксированных значений параметра S, PD-sbox с размерностью входа и выхода 4 бит при $K = 4$ по качеству дифференциальных свойств подобен операциям подстановки алгоритма AES, при $K = 5$ превосходит – максимальное отклонение $R_P(\Delta X, \Delta Y)$ от идеального значения в два раза меньше, чем для подстановок AES. Потенциально можно достичь крайне низких значений отклонения $R_P(\Delta X, \Delta Y)$ от идеального значения при приемлемых вычислительных затратах [31].

При конкретном значении состоянии S эквивалентная операция подстановки с большой вероятностью будет невзаимооднозначной, следовательно, не может обладать идеальными дифференциальными

свойствами. Однако, при усреднении дифференциальных характеристик по всем возможным порождаемым подстановкам статистические отклонения дифференциалов для отдельных эквивалентных замен взаимно компенсируются, так как выходные значения PD-sbox равновероятны при равновероятном изменении значений состояния S. Попытка анализа отдельных эелементов PD-sbox (для обхода усреднения характеристик) затруднена структурой псевдо-динамической операции подстановки – внутреннее состояние значительно превосходит размерность входов и выходов, что значительно усложняет дифференциальный криптоанализ. PD-sbox, состоящие из взаимооднозначных фиксированных операций подстановки, обладают идеальными дифференциальными свойствами при динамическом равновероятном изменении значений состояний S [31].

На основе результатов следует сформировать дополнительные требования к фиксированным операциям подстановки, составляющим PD-sbox:

– фиксированные подстановки должны быть взаимооднозначными;

– иметь минимально возможные отклонения дифференциальных характеристик от идеального значения;

– обладать достаточной нелинейностью с целью разрушения статистических связей между выходными значениями для эффективного изменения значений состояния S.

Резюме. Псевдо-динамические операции подстановки PD-sbox способны значительно снизить отклонения коэффициента распространения дифференциалов от идеального значения, в частности, путём увеличения числа составляющих её фиксированных замен при равновероятности выбора значения состояния S, задающего конкретную порождаемую операцию подстановки. Данный эффект, в отличие от динамических таблиц подстановок, достигается без значительного увеличения используемых вычислительных ресурсов [31].

1.5. Описание метода автоматизированного поиска криптографических характеристик с использованием SMT решателей и библиотеки CASCADA

На ранних этапах диссертационного исследования осуществлялся поиск и анализ программного обеспечения, предназначенного для автоматизированного определения криптографических свойств как фиксированных sbox, так и, в перспективе, псевдо-динамических подстановок. После рассмотрения ряда программных инструментов, в частности SET, S-Box Analyzer и S-Box Inspector, были сделаны выводы о том, что указанные утилиты оценивают свойства только одного sbox, переданного решению для анализа, а не набора криптографических подстановок. Следовательно, для работы с крупными массивами подстановок, что необходимо для подбора оптимальных вариантов замен, потребуется модификация кода. Помимо этого, рассмотренные продукты не используют многопоточность современных процессоров или вычислительные мощности графических ускорителей, что существенно замедляет процесс анализа [40].

Для решения указанной проблемы было разработано собственное решение, совместимого с многопоточными системами. Первая версия программного обеспечения рассмотрена в [41]. Цель работы заключалась в разработке и тестировании программного инструмента, использующего современные многопоточные процессоры для анализа нелинейных характеристик криптографических подстановок. На его основе было разработано программное обеспечение, предназначенное для анализа криптографических свойств (дифференциальные свойства, линейные свойства, алгебраический иммунитет, количество степеней свободы) псевдо-динамических операций подстановки на основе ARX-конструкций [42]. Однако, для наиболее наглядного сравнения свойств синтезируемых псевдо-динамических подстановок требовался иной инструмент, в частности – CASCADA, широко распространённый фреймворк для анализа криптографических характеристик, использующий SAT-решатели.

Автоматизированные методы поиска криптографических характеристик, основанные на задачах выполнимости формул в теориях (SMT – Satisfiability Modulo Theories), широко используются для оценки безопасности блочных шифров, псевдослучайных функций и псевдослучайных перестановок от атак с распознаванием. Эти методы обеспечивают систематическую и общую методологию, однако большинство их программных реализаций ограничены небольшим набором криптоалгоритмов и атак.

CASCADA – библиотека Python 3 с открытым исходным кодом для оценки безопасности криптографических примитивов, блочных шифров, PRF и PRP, против атак с распознаванием с помощью бит-векторных SMT решателей. CASCADA реализует структуру свойств бит-вектора и ряд методов автоматизированного поиска на основе SMT для оценки безопасности против различных видов криптоанализа [43].

Бит-векторное выражение – это бит-векторная константа, бит-векторная переменная или бит-векторная операция с бит-векторными выражениями в качестве входных данных. Константы интерпретируются как целые числа без знака, n-битный вектор $x = b_{n-1} \cdots b_1 b_0$ обозначает неотрицательное целое число $b_0 + 2b_1 + \cdots + 2^{n-1}b_{n-1}$, b_i также обозначается $x[i]$, где $x[0] = b_0$ обозначает наименее значимый бит (LSB), а $x[n-1] = b_{n-1}$ обозначает наиболее значимый бит (MSB). Представлены следующие бит-векторные операции и обозначения: объединение и извлечение битовых векторов; побитовые логические операции: отрицание ¬, конъюнкция ∧, дизъюнкция ∨ и исключающее или (XOR) $\oplus$; операции логического сдвига: сдвиг влево $\ll$, сдвиг вправо $\gg$, циклический сдвиг влево $<<<$ и циклический сдвиг вправо $>>>$; арифметические операции: модульное сложение $\boxplus$, модульное вычитание $\boxminus$, модульное умножение $\boxtimes$, операция усеченного деления без знака ÷ и операция остатка без знака (по модулю) %; реляционные операции: =, <, >, ≤ и ≥; оператор if-then-else $Ite(b, x, y)$, возвращающий x, если b равно биту 0, и в противном случае возвращающий y.

Бит-векторная формула или ограничение – это выражение, возвращающее один бит, где 0 обозначает значение False, а 1 обозначает True. Бит-векторная формула выполнима, если существует присвоение переменных, которое делает формулу истинной [43].

Задача выполнимости формул в теориях (SMT) относится к проблеме определения того, является ли формула первого порядка выполнимой относительно некоторой логической теории. Задачи SMT можно рассматривать как обобщение задач SAT (Задача выполнимости булевых формул); последние выражаются в пропозициональной логике, а задачи SMT даются в более обширной логике, такой как теория целых чисел или теория битовых векторов.

Программные средства, которые определяют выполнимость SMT-задач, называются SMT-решателями. За последние два десятилетия популярность SMT-решателей возросла благодаря технологическому прогрессу и промышленному применению в разработке программного обеспечения, оптимизации и многих других областях. Вдобавок ко всему, в настоящее время доступно множество самых современных SMT-решателей с открытым исходным кодом, таких как Boolector или STP.

Решатели SMT могут не только определить выполнимость SMT-задачи, но и найти значения переменных, удовлетворяющих задаче. Эта функция позволяет использовать SMT-решатели в задачах поиска. Используя SMT-решатель, поддерживающий бит-векторную теорию, возможно проверить, выполнима ли задача, и в этом случае найти значение k, которое составляет выражение $f_k(0,0) = (0,0)$ [43].

Инструмент CASCADA основан на ArxPy, инструменте для поиска дифференциальных характеристик и невыполнимо-дифференциальных свойств шифров ARX. Однако, в то время как ArxPy ограничивается дифференциальным анализом (связанного ключа), RX невыполнимо-дифференциальным криптоанализом (связанного ключа), CASCADA

реализует структуру свойств бит-вектора, новые автоматизированные методы и множество функциональных возможностей и улучшений.

CASCADA реализует дифференциальные модели XOR, дифференциальные модели RX и линейные модели многих бит-векторных операций, а также реализует слабые модели и модели на основе ветвей и модель свойств, основанную на таблицах весов. В результате CASCADA может выполнять поиск (по связанному ключу) XOR-дифференциальных характеристик, RX-дифференциальных характеристик и линейных свойств, а также может выполнять поиск (по связанному ключу) XOR-невозможных дифференциалов, RX-невозможных дифференциалов и линейных приближений с нулевой корреляцией.

По сравнению с ArxPy, интерфейс в CASCADA улучшен для поддержки не только алгоритмов на основе ARX-операций, но и других шифров и примитивов, документация расширена таким образом, что каждая функция и класс Python содержат подробную документацию с примерами использования. CASCADA включает в себя полный набор тестов для проверки свойств на случайных входных данных.

Алгоритм запуска одного из методов автоматического поиска с помощью CASCADA выглядит следующим образом. Сначала пользователь реализует примитив, следуя интерфейсу, предоставляемому CASCADA; он также может выбрать один из множества уже реализованных примитивов. Затем, если модель свойств операции примитива не предоставляется CASCADA, пользователь может либо реализовать модель свойств, либо просто использовать слабую модель на основе ветвей или таблиц. Наконец, пользователь выбирает метод поиска и его параметры (например, свойство бит-вектора, решатель SMT, дополнительные ограничения и так далее) и запускает поиск. В процессе поиска CASCADA генерирует характеристическую модель из реализации примитива на Python, кодирует задачи SMT и решает их путем запроса внешнего SMT-решателя. Шаги, представленные на рисунке 1.12 выполняются CASCADA. Таким образом,

использование CASCADA не требует глубокого понимания работы SMT-решателей, поскольку функция автоматически обрабатывается CASCADA. Время поиска определяется затратами на решение SMT-задач, а шаги, выполняемые CASCADA, приводят к незначительным накладным временным расходам [43].

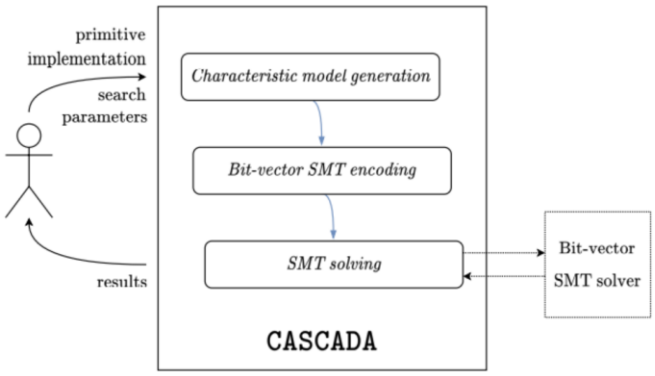


Рисунок 1.12 – Основные шаги, выполняемые CASCADA

Библиотека CASCADA имеет модульную конструкцию со свободной связью: модуль бит-вектора, модуль примитива, модули свойств и модуль SMT, так что каждый модуль может использоваться и расширяться независимо.

Резюме. Описан метод автоматической оценки свойств криптографических примитивов, криптоалгоритмов, псевдослучайных функций и псевдослучайных перестановок с использованием SMT решателей. Задача выполнимости формул в теориях (SMT) относятся к проблеме определения того, является ли формула первого порядка выполнимой относительно некоторой логической теории. Задачи SMT рассматриваются как обобщение задач SAT (Задача выполнимости булевых формул). Инструмент CASCADA позволяет осуществлять анализ криптографических характеристик и призван не только облегчить разработчикам и криптоаналитикам оценку безопасности криптографических примитивов, но и помочь в дальнейших исследованиях автоматизированных методов.

1.6. Постановка актуальной научной задачи и формулировка частных задач

Результаты анализа демонстрируют необходимость разработки структуры псевдо-динамической операции подстановки, удовлетворяющей множеству взаимоисключающих требований, в частности, линейным и разностным характеристикам, не уступающим фиксированным операциям подстановки той же размерности, а также эффективности и скорости программной реализации PD-sbox и криптографических преобразований на её основе.

Существующие подходы к синтезу и применению динамических операций подстановки не позволяют одновременно обеспечить стойкость, минимизацию затрачиваемых ресурсов и скорость программной реализации псевдослучайных функций на их основе, сопоставимую с псевдослучайными функциями на основе фиксированных подстановок или иных фиксированных преобразований. В отличие от этого метод синтеза псевдо-динамических подстановок на основе ARX-функций, позволяет получать преобразования, удовлетворяющие требованиям по криптографическим свойствам, затрачиваемым ресурсам и скорости программной реализации криптографических преобразований.

В связи с вышесказанным возникает актуальная научная задача разработки и исследования метода синтеза псевдо-динамических подстановок на основе ARX-функций, удовлетворяющих широкому спектру противоречивых требований по стойкости к криптоанализу и затрачиваемым ресурсам при программной реализации криптографических преобразований.

Целью диссертационного исследования является минимизация затрачиваемых ресурсов программной реализации криптографических преобразований при обеспечении заданных криптографических свойств, посредством разработки метода синтеза псевдо-динамических подстановок на основе ARX-функций.

Достижение поставленной цели предусматривает решение **частных задач:**

1. Анализ существующих подходов к синтезу псевдо-динамических операций подстановки.

2. Синтез структуры псевдо-динамической операции подстановки, удовлетворяющей широкому спектру противоречивых требований, посредством разработки метода синтеза псевдо-динамических подстановок на основе ARX-функций.

3. Анализ синтезированной псевдо-динамической функции PD-sbox-ARX-32 и её программной реализации на малоресурсных процессорах.

Объект исследования – криптографические операции подстановки, являющиеся составным элементом множества блочных шифров.

Предмет исследования – синтез и исследование псевдо-динамических операций подстановки, удовлетворяющих широкому спектру противоречивых требований по стойкости к разностному криптоанализу, а также затрачиваемым ресурсам при программной реализации криптографических преобразований.

Методы исследования: статистический криптоанализ с использованием SMT/SAT решателей, численные методы для оценки свойств псевдо-динамических подстановок, вычислительный эксперимент по определению криптографических свойств ARX-функций и псевдо-динамической функции PD-sbox-ARX-32.

Теоретическая значимость результатов исследования состоит в развитии перспективного научного направления синтеза и применения псевдо-динамических подстановок на основе ARX-функций, удовлетворяющих широкому спектру противоречивых требований по стойкости к криптоанализу и затрачиваемым ресурсам при программной реализации криптографических преобразований.

2. СИНТЕЗ ПСЕВДО-ДИНАМИЧЕСКОЙ ОПЕРАЦИИ ПОДСТАНОВКИ НА ОСНОВЕ ARX-ФУНКЦИЙ

Актуальным направлением современной криптографии является создание и исследование криптоалгоритмов на основе ARX-функций. ARX-конструкции наиболее эффективно используют возможности современных процессоров и обеспечивают высокую скорость преобразования информации.

Развитием структуры псевдо-динамических операций подстановки PD-sbox является применение в их составе ARX-конструкций в качестве основного нелинейного элемента, на замену фиксированным подстановкам. Несмотря на слабые криптографические свойства ARX-функций, они позволяют значительно уменьшить затраты ресурсов при программной реализации и получить криптографические свойства конструкции, аналогичные фиксированным операциям подстановки той же размерности.

Следует отметить, что для синтеза оптимальной ARX-функции необходимо осуществить подбор значений сдвига для наиболее эффективного использования вычислительных возможностей современных процессоров.

2.1. Описание и подбор ARX-функций, адаптированных для работы в составе PD-sbox

ARX-конструкция – это функция, состоящая из трёх типов операций: сложение по модулю, циклический сдвиг, сложение по модулю 2 (XOR). ARX-функции могут являться составной частью псевдослучайных функций и криптоалгоритмов. В частности, на их основе построены такие шифры, как ChaCha20, Speck, XXTEA, и BLAKE [44]. Пример типовой ARX-конструкции, представлен на рисунке 2.1.

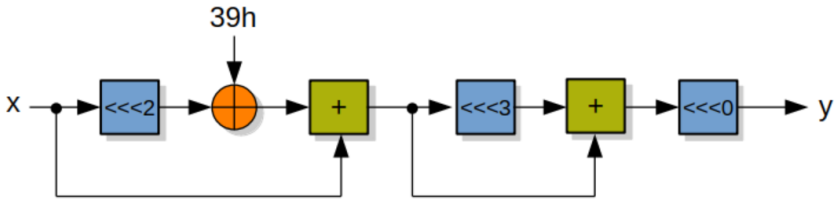


Рисунок 2.1 – Типовая ARX-функция

Особенностями псевдослучайных функций и криптоалгоритмов, построенных на основе ARX-конструкций являются высокая скорость преобразований, а также малое потребление ресурсов как при программной, так и при аппаратной реализации. Простота конструкции обеспечивает работоспособность подобных алгоритмов в режиме реального времени (отсутствие задержек на sbox), что значительно уменьшает вероятность применения атак по времени.

Для раскрытия возможностей ARX-функций в составе псевдо-динамических операций подстановки предложена ARX-конструкция, позволяющая эффективно использовать ресурсы и особенности современных процессоров и аппаратных платформ, в частности AVX-инструкции (Advanced Vector Extensions). Основными критериями синтеза ARX-функций являются использование сдвигов, кратных 8 бит, и параллельность выполнения представленных в ней операций, что способствует оптимальному использованию AVX-инструкций.

Выражение, описывающее 64-битную ARX-функцию (значения a и b являются 32-битными):

$$a_2 = \big((a_1 + (a_1 \ll t_0) \oplus b_1) \ll t_2\big) \oplus const_0, \quad (31)$$

$$b_2 = \big((b_1 + (b_1 \ll t_1) \oplus a_1) \ll t_3\big) \oplus const_1, \quad (32)$$

$$a_3 = (a_2 + \big((a_2 \ll t_4) \oplus b_2\big) \ll t_6), \quad (33)$$

$$b_3 = (b_2 + \big((b_2 \ll t_5) \oplus a_2\big) \ll t_7), \quad (34)$$

где $\oplus$ – сложение по модулю 2, $a \ll t$ – циклический сдвиг на t бит влево в двоичном слове a, a_1 – младшие 32 бит входного 64-битного значения, b_1 –

старшие 32 бит входного 64-битного значения, a_3 и b_3 – младшие и старшие 32 бит выходного 64-битного значения, $t_0 \dots t_7$ – значения циклических сдвигов, задающие конкретную ARX-функцию, $const$ – 32-битные значения констант, задающие конкретную ARX-функцию.

Значения описанных параметров представлены в таблице 2.1.

Таблица 2.1 – Значения параметров ARX-функций

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
funcARX0:	8	16	16	8	8	16	0	0
funcARX1:	8	16	8	16	16	8	8	8
funcARX2:	16	8	8	16	8	16	16	16
funcARX3:	16	8	16	8	16	8	24	24

Структура подобранной ARX-функции представлена на рисунке 2.2.

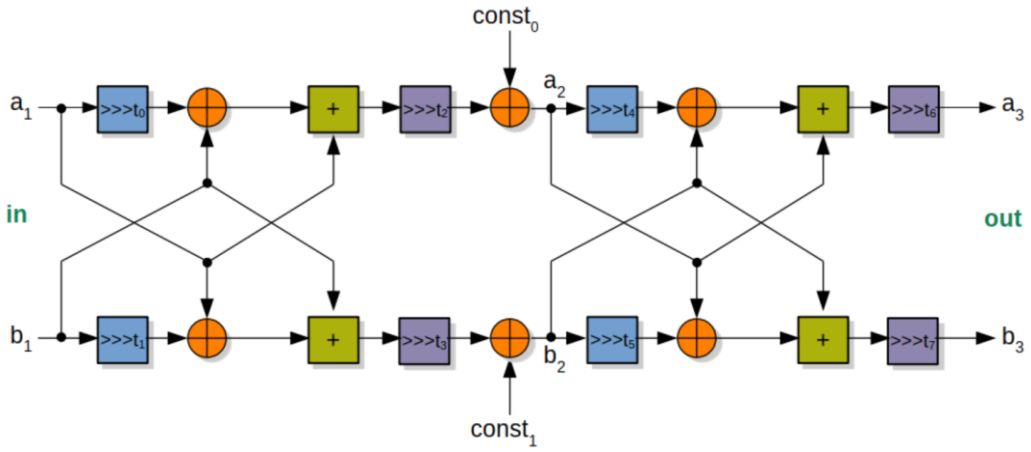


Рисунок 2.2 – Структура ARX-функции

На базе представленных ARX-конструкций построена 64-битная псевдо-динамическая операция подстановки PD-sbox_4x64x64, включающая в свой состав 4 ARX-функции. PD-sbox_4x64x64 представлена на рисунке 2.3.

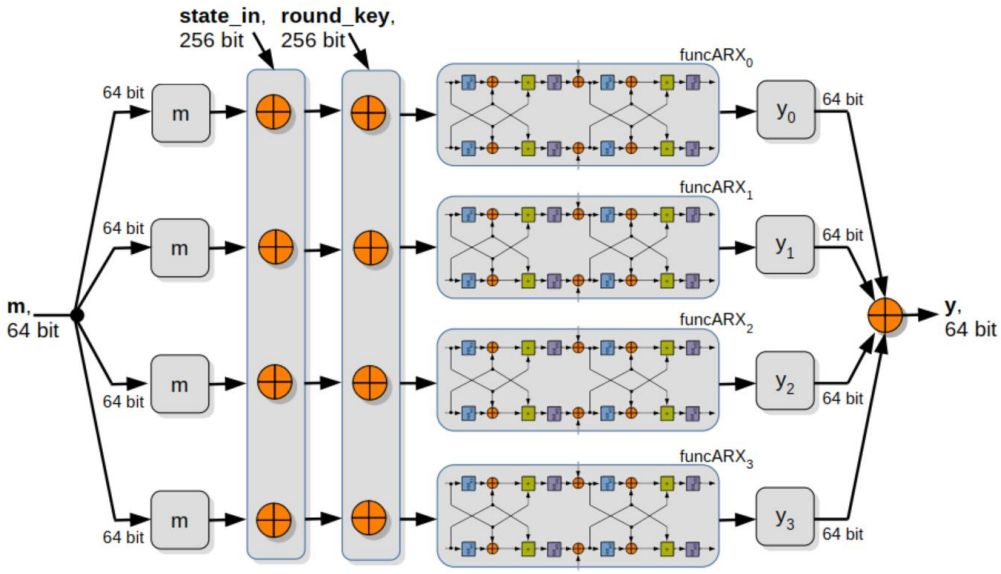


Рисунок 2.3 – PD-sbox_4x64x64

Размер предполагаемого раундового ключа – 256 бит, размер управляющего состояния для изменения порождаемых операций подстановки – 256 бит. Суммарное значение внутреннего состояния – 512 бит [45].

Стоит отметить, что использование псевдо-динамических подстановок на основе ARX-функций в PRF семейства pCollapser позволяет получить из набора 4 ARX-функций с предельно низкими криптографическими свойствами, качественную нелинейную функцию, что подтверждает правильность концепции псевдо-динамических подстановок PD-sbox [34].

2.2. Анализ структуры sbox Alzette

Рассмотрим криптографические преобразования, построенные на основе ARX-функций. В работе [46] представлен sbox размерностью 64 бит, реализованный на основе ARX-конструкций – Alzette. Данное преобразование может быть вычислено с использованием 12 инструкций на современных процессорах, а параллельная реализация sbox Alzette может использовать векторные (SIMD) инструкции. За одну итерацию Alzette достигает разностных и линейных свойств, сравнимых со свойствами sbox AES [46]. Alzette обладает следующим преимуществом – использование операций размерностью 32 бит, следовательно, согласно [47], его эффективная

реализация возможна на множестве различных архитектур, благодаря использованию регистров сдвига (barrel shift registers), если они доступны, а также благодаря подобранным значениям операций циклического сдвига.

Рассмотрим структуру Alzette, представленную на рисунке 2.4. Функция параметризована константой c размерностью 32 бит, используемой 4 раза. На вход Alzette поступает два слова размерностью 32 бит. Далее над каждым из них выполняются операции циклического сдвига, сложения по модулю 2^{32}, а также XOR.

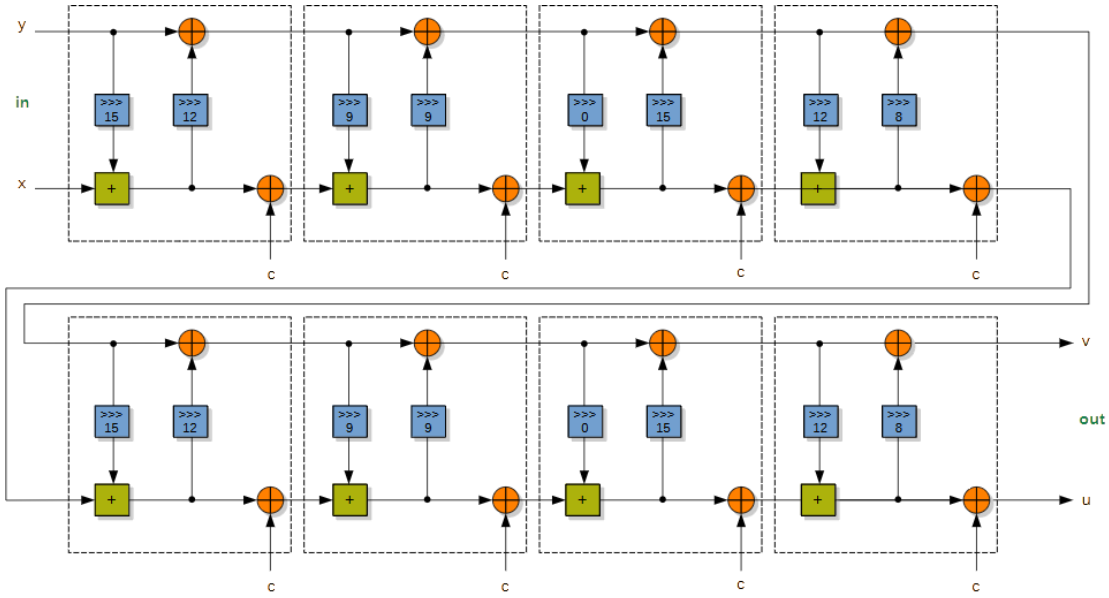


Рисунок 2.4 – Структура sbox Alzette (8 раундов)

При использовании Alzette в составе преобразования Sparkle, используются следующие константы: $c_0 = b7e15162$, $c_1 = bf715880$, $c_2 = 38b4da56$, $c_3 = 324e7738$, $c_4 = bb1185eb$, $c_5 = 4f7c7b57$, $c_6 = cfbfa1c8$, $c_7 = c2b3293d$. Sparkle – это семейство криптографических преобразований размерностью 256, 384 или 512 бит. Sparkle включает в свой состав операции сложения по модулю слова, циклического сдвига и XOR (ARX-конструкции). Основой перестановки являются sbox на основе ARX-функций Alzette [48].

Таким образом, применение ARX-операций позволило разработчикам Alzette создать функцию (блок замены) размерностью 64 бит. Это значительно больше, чем типовая размерность блоков замены в криптографических преобразованиях. Следовательно, прямой расчёт криптографических свойств

(например, таблиц распределения разностей DDT и линейных приближений LAT) для sbox Alzette крайне затруднителен или вовсе невозможен, так как требует недоступного объёма вычислительных ресурсов. Поэтому, авторы Alzette применяли SAT-решатели для определения её криптографических свойств [46].

Применительно к sbox Alzette её авторы используют термин “раунд” (итерация), по аналогии с блочными криптографическими преобразованиями (например, как в криптоалгоритме Speck64, структура которого представлена на рисунке 2.5). В таблице 2.2 приведены границы разностных и линейных свойств для версий Alzette с использованием от 1 до 12 раундов. Основная версия Alzette предполагает 4 раунда преобразования [48].

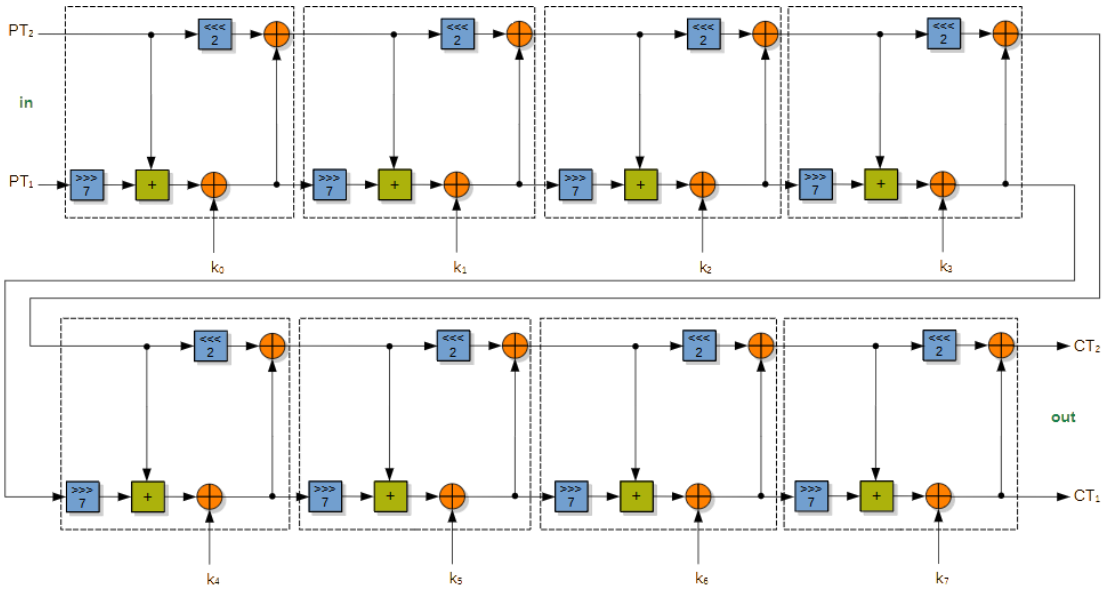


Рисунок 2.5 – Структура Speck32 (8 раундов)

Таблица 2.2 – Нижние границы для разностных и линейных свойств sbox Alzette

$(r_0, r_1, r_2, r_3, s_0, s_1, s_2, s_3)$	1	2	3	4	5	6	7	8	9	10	11	12
$(31, 17, 0, 24, 24, 17, 31, 16)$	0 0	1 0	2 1	6 2	10 5	18 8	≥24 13(11.64)	≥32 17(15.79)	≥36 –	≥42 –	≥46 –	≥52 –
$(17, 0, 24, 31, 17, 31, 16, 24)$	0 0	1 0	2 1	6 2	10 5	17 9	≥25 13	≥31 16	≥37 –	≥41 –	≥47 –	– –

Продолжение таблицы 2.2

$(r_0, r_1, r_2, r_3, s_0, s_1, s_2, s_3)$	1	2	3	4	5	6	7	8	9	10	11	12
$(0, 24, 31, 17, 31, 16, 24, 17)$	0 0	1 0	2 1	6 2	10 6	18 8	≥24 13	≥32 15	≥36 –	≥42 –	– –	– –
$(24, 31, 17, 0, 16, 24, 17, 31)$	0 0	1 0	2 1	6 2	10 5	17 9	≥25 12	≥31 16	≥37 –	– –	– –	– –
Speck64	0 0	1 0	3 1	6 3	10 6	15 9	21 13	29 17	≥32 19	– 21	– 24	– 27

Для каждого смещения первая строка демонстрирует $\log_2 p$, где p – максимальная ожидаемая вероятность дифференциального следа для дифференциального случая. Вторая строка демонстрирует $\log_2 c$, где c – максимальная ожидаемая абсолютная линейная корреляция следа для линейного случая. Значение, указанное в скобках, соответствует максимальной абсолютной корреляции линейной оболочки с учетом кластеризации, полученной экспериментальной проверкой.

Как и в блочных преобразованиях, авторы Alzette используют широко распространённый метод поиска разностных и линейных характеристик – когда определяются разностные и линейные свойства отдельных раундов и результирующие свойства всего преобразования определяются через «принцип накопления» (piling-up principle) [49]. Данный принцип предполагает, что входные значения каждого раунда являются статистически независимыми.

Стоит отметить одну важную особенность – в Alzette между «раундами» не используется добавление раундовых ключей, что, предположительно, существенно нарушает «принцип накопления». Однако, это не помешало авторам Alzette получить валидные криптографические свойства используя SAT-решатели и применяя метод Мацуи для поитерационного поиска разностных и линейных характеристик Alzette [48].

2.3. Первичный анализ криптографических свойств псевдо-динамических подстановок на основе подобранных ARX-функций

В ходе диссертационного исследования на ранних этапах осуществлялся анализ линейных свойств эквивалентных подстановок, в частности, полученных с использованием псевдо-динамических подстановок на основе ARX-функций. В работе [50] результаты демонстрируют, что эквивалентные замены, сгенерированные псевдо-динамической операцией подстановки на базе ARX-конструкций обладают линейными характеристиками, сопоставимыми со случайно-сгенерированными 8-битными небиективными sbox аналогичной размерности, несмотря на применение ARX-функций, обладающих неудовлетворительными криптографическими свойствами. В работе [51] выполнен анализ количества активных ARX-функций для мини-версии PRF pCollapser-ARX, результаты исследования демонстрируют, что минимальная доля активных ARX-функций (от их общего количества) увеличивается с каждым раундом и достигает значения 0,594 для 4 раунда. Для примера, минимальная доля активных S-боксов для криптоалгоритма Present составляет 0,125 для 31 раунда. Результаты исследования показывают, что несмотря на скромные нелинейные свойства исследуемых ARX-функций, их объединение в псевдо-динамические подстановки достаточно для пораундового наращивания сложности преобразования мини-версии pCollapserARX. Стоит отметить отсутствие коллизий выходных значений для всех опробованных разностей. (Наличие коллизий проявилось бы в отсутствии увеличения активных ARX-функций после определённого раунда, что привело бы к резкому снижению сложности преобразования для соответствующих входных разностей) [51].

Далее, при учёте доступных вычислительных возможностей, проведены экспериментальные исследования линейных и дифференциальных свойств для псевдо-динамических подстановок на основе ARX-функций размерностью 12 и 24 бит. В данном случае, таблицы линейных

аппроксимаций (LAT) и таблицы распределения дифференциалов (DDT) рассчитывались частично, для небольшого количества элементов этих таблиц.

Это вызвано тем, что целью был не поиск максимальных значений в таблицах LAT и DDT, а демонстрация возможностей получаемых псевдо-динамических операций подстановки – когда объединение ARX-функций, имеющих откровенно слабые криптографические свойства, в структуру псевдо-динамической замены позволяет получать свойства порождаемых эквивалентных подстановок, близкие к свойствам случайно сформированных операций подстановки аналогичной размерности.

Предполагается, что свойства ARX-функций и получаемых псевдо-динамических подстановок будут сохраняться при масштабировании до 64 бит. Результаты вычислений показаны в таблице 2.3 [52].

Таблица 2.3 – Результаты исследования свойств псевдо-динамических подстановок ARX

Тип подстановки	**#tested sbox**	**max bias (LAT)**	**max N_D (DDT)**
n = 12 bit, *ARX-function with: 4 add, 8 rol*			
Random sbox 12x12 (non-bijective)	100	0.0432129	20
ARX_2x6_4a8r	**4**	**0.127**	**268**
Equivalent sbox 12x12	100	0.0507812	20
Dynamic sbox 48x12	**100**	**0.0035986**	**1.84**
– «» –	**1000**	**0.0001137**	**1.24**
n = 24 bit, *ARX-function with: 4 add, 8 rol* (partial computations)			
Random sbox 24x24 (non-bijective)	100	0.000479341	20
ARX_2x12_4a8r	**4**	**0.0012017**	**74918**
Equivalent sbox 24x24	100	0.0006589	20
Dynamic sbox 96x24	**100**	**0.0000498**	**2.56**
– «» –	**1000**	**0.0000098**	**2.174**
n = 64 bit (estimated values)			
Equivalent sbox 64x64	-	~ random_sbox64	~ random_sbox64
Dynamic sbox 256x64	**-**	**~ ideal bias**	**~ ideal N_D**

Выполнен анализ дифференциальных свойств эквивалентных операций подстановки размерностью 24 бит, сгенерированных PD-sbox-ARX, а также случайно-сформированных sbox аналогичной размерности.

Проанализированы 100 эквивалентных подстановок PD-sbox-ARX и случайно-сформированных sbox. Получена гистограмма распределения усреднённых значений в таблице DDT, представленная в таблице 2.4. Следует обратить внимание на то, что использован частичный расчёт значений таблицы DDT [52].

Таблица 2.4 – Гистограмма распределения усреднённых значений в таблице DDT, 100 подстановок (был использован частичный расчёт таблицы)

PD-sbox-ARX_24x24			Random sbox 24x24		
[minval = 0.16, maxval = 2.56, step = 0.12, counted values = 469762020]			[minval = 0.3, maxval = 1.94, step = 0.082, counted values = 469762020]		
num bin	hist_x	hist_y	num bin	hist_x	hist_y
0	0.160	2	0	0.300	34
1	0.280	560	1	0.382	1449
2	0.400	36006	2	0.464	32193
3	0.520	1167654	3	0.546	397440
4	0.640	7027328	4	0.628	2847586
5	0.760	39649383	5	0.710	12671131
6	0.880	107018082	6	0.792	36791751
7	1.000	168913627	7	0.874	72612464
8	1.120	95997949	8	0.956	100691648
9	1.240	37146470	9	1.038	100875924
10	1.360	9727624	10	1.120	88173452
11	1.480	2250120	11	1.202	34756780
12	1.600	539387	12	1.284	14055495
13	1.720	210943	13	1.366	4461131
14	1.840	58472	14	1.448	1123076
15	1.960	14516	15	1.530	227248
16	2.080	3167	16	1.612	37494
17	2.200	599	17	1.694	5070
18	2.320	116	18	1.776	603
19	2.440	13	19	1.858	45
20	2.560	2	20	1.940	6

2.4. Метод синтеза псевдо-динамической функции PD-sbox-ARX-32

Разработан универсальный метод синтеза PD-sbox-ARX, для микроконтроллеров семейства AVR. Доказательство эффективности

предложенного метода приводится для микроконтроллера архитектуры AVR – ATmega328P и конкретной реализации псевдо-динамической функции PD-sbox-ARX-32. Математическое обоснование представлено в работе [53]:

1. Эвристический выбор структуры ARX-функции с учётом возможных особенностей программной и аппаратной реализаций;

2. Начальное заполнение параметров циклических сдвигов ARX-функций (всего по 8 значений на 4 функции) значением 8;

3. Последовательный выбор каждого параметра ARX-функции, замена его на случайное значение из допустимого диапазона (от 0 до 15), проверка криптографических свойств (вес разностных и линейных характеристик) и ожидаемого количества затрачиваемых ассемблерных инструкций для полученной версии ARX-функции. После обхода всех параметров первой ARX-функции выбирается наилучшая версия (при её наличии), которая заменяет исходную. Далее осуществляется переход к следующей ARX-функции для выполнения аналогичных действий;

4. Действия из пункта 3 повторяются до момента отсутствия улучшений в свойствах ARX-функций;

5. Формирование при помощи пунктов 2 и 3 набора наиболее удачных ARX-функций;

6. Выбор из набора наиболее удачных ARX-функций варианта с наименьшим количеством затрачиваемых ассемблерных инструкций для микроконтроллеров архитектуры AVR. В таблице 2.4 представлено количество ассемблерных инструкций для реализации 16-битовых операций ROL для 12 отобранных параметров PD-sbox-ARX-32.

Следует обратить внимание на следующее: по пункту 2 экспериментальные исследования показали, что если сразу задать «удачный» вариант значений циклического сдвига (для 16-битных сдвигов это значение равно 8), то значительно увеличивается вероятность того, что эти значения будут в результирующей ARX-функции после операций синтеза; по пункту 5 экспериментальные исследования показали, что предложенный пошаговый

подбор параметров позволяет получать PD-sbox-ARX с достаточно близкими к 8-раундовым преобразованиям Speck32 и miniAlzette32 криптографическими свойствами. При синтезе 100 PD-sbox-ARX 73 варианта имели вес разностных характеристик Wd равный 32 и вес линейных характеристик Wl, равный 13 и 14. Такие характеристики очень близки 8-раундовым преобразованиям Speck32 и miniAlzette32. Поэтому варианты с более худшими характеристиками исключаются из набора [53].

В результате получены 12 параметров, представленных в таблице 2.5, при этом наименьшее количество ассемблерных инструкций для архитектуры AVR составило $N = 360$, что в 1,4 раза и 1,7 раза больше, чем для наихудшего и наилучшего вариантов синтеза соответственно, для которых $N = 256$ и $N = 217$.

Таблица 2.5 – Количество ассемблерных инструкций для реализации 16-битовых операций ROL для 12 отобранных параметров PD-sbox-ARX-32

№	**Архитектура микроконтроллера**		
	AVR	mips64	ARM
1	242	150	73
2	248		
3	256		
4	222		
5	242		
6	248		
7	240		
8	231		
9	**217**		
10	234		
11	234		
12	248		

В таблице 2.6 приведено сравнение свойств лучшей синтезированной PD-sbox-ARX-32 со свойствами 8-итерационной 32-битной Alzette-подобной структуры и 8-итерационным 32-битным преобразованием из блочного криптоалгоритма Speck32 [53].

Таблица 2.6 – Количество ассемблерных инструкций для реализации 16-битовой операций ROL

Преобразование	**Архитектура микроконтроллера**			**Криптографические свойства**			
	AVR	mips64	ARM	Wd	Wde	Wl	Wle
miniAlzette32 (8 раундов)	154	70	42	27	~27	13	~13
Speck32 (8 раундов)	240	80	48	24	~24	12	~12
PD-sbox-ARX-32	**217**	**150**	**73**	**32**	**~26**	**13**	**~13**

2.5. Выводы

Синтезирована структура псевдо-динамической операции подстановки на основе ARX-функций. Исследования демонстрируют, что объединение ARX-функций, имеющих откровенно слабые криптографические свойства, в структуру псевдо-динамической подстановки позволяет получать свойства эквивалентных подстановок, близкие к свойствам случайно сформированных подстановок аналогичной размерности. PD-sbox-ARX содержит простые операции и имеет заложенные возможности параллелизации обработки данных, что позволяет делать эффективные программные и аппаратные реализации для различных процессоров и аппаратных платформ.

Основное назначение псевдо-динамических операций подстановки на основе ARX-функций – применение в высокопроизводительных PRF в режимах, где не требуется обратимость преобразования: AEAD, CTR, Sponge-конструкции и др. Для задач легковесной криптографии PD-sbox-ARX легко масштабируется путём изменения размерности слов.

Предложен метод синтеза псевдо-динамической функции PD-sbox-ARX-32, который позволяет получать PD-sbox-ARX с достаточно близкими к 8-раундовым преобразованиям Speck32 и miniAlzette32 криптографическими свойствами. При синтезе 100 PD-sbox-ARX 73 варианта имели вес разностных характеристик Wd равный 32 и вес линейных характеристик Wl, равный 13 и 14 [53].

Результаты исследований, представленные в главе, опубликованы в статьях [53, 34, 50, 51], цитируемых в РИНЦ и в рецензируемых журналах, входящих в перечень ВАК РФ. Результаты исследований апробированы на ежегодной научно-практической конференции «РусКрипто» (РусКрипто'2022), научный доклад «Высокопроизводительная псевдослучайная функция pCollapserARX256-32x2» (Москва, 22 – 25 марта 2022 г.), а также на VIII Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности» (4 – 9 апреля 2022 г.) и IX Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности» (10 – 15 апреля 2023 г.).

Программный продукт для ЭВМ «Программа анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций» разработана в процессе научных исследований на кафедре ИБТКС ЮФУ, что подтверждено актом о внедрении, представленном в приложении В (№ 2024610931 В Реестре 16.01.2024 (Приложение D).

3. АНАЛИЗ ПСЕВДО-ДИНАМИЧЕСКОЙ ФУНКЦИИ PD-SBOX-ARX-32 И ЕЁ ПРОГРАММНОЙ РЕАЛИЗАЦИИ НА МАЛОРЕСУРСНЫХ ПРОЦЕССОРАХ

Рассмотрим анализ криптографических характеристик синтезированной псевдо-динамической функции PD-sbox-ARX-32, в частности разностных и линейных, а также сравним их с аналогичными параметрами миниверсии sbox Alzette, размерностью 32 бита.

В силу размерности и сложности анализируемых конструкций, для поиска разностных и линейных свойств использован метод поиска криптографических характеристик, использующий SAT-решатели, в частности – фреймворк CASCADA [43]. Создатели данного фреймворка особое внимание уделили анализу ARX-функций. В настоящее время SAT-решатели являются одним из основных методов поиска и валидации криптографических свойств и характеристик криптографических преобразований [43, 54–57].

Свойства, используемые в разностном криптоанализе, являются разностями (α, β) по функции шифрования E_k с высокой ожидаемой разностной вероятностью. При наличии разностей (α, β) по f его разностная вероятность определяется как:

$$\#\{x: f(x\Delta\,\alpha)\nabla f(x) = \beta\}/2^n\,, \tag{35}$$

где оператор ∇ вычисляет разность пары значений (x, x'), а оператор Δ принимает в качестве входных данных значение x и разность α и выводит такое значение x', что пара (x, x') имеет разность α [57].

Ожидаемая разностная вероятность p — это разностная вероятность, усредненная по ключевому пространству K:

$$p = \frac{1}{|K|}\sum_{k\in K}\#\{x: f(x\Delta\alpha)\nabla\mathrm{f(x)} = \beta\}/2^n, \tag{36}$$

а сложность разностного криптоанализа составляет $O(1/p)$.

Свойство (α, β) над функцией f является действительным, если его вероятность распространения не равна нулю. В этом случае мы определяем вес распространения (α, β) как отрицательный двоичный логарифм его вероятности распространения:

$$PW_f(\alpha, \beta) = -\log_2\Big(PP_f(\alpha, \beta)\Big). \quad (37)$$

Результат работы CASCADA – веса найденных оптимальных криптографических характеристик:

$$w = -\log_2 P(.), \quad (38)$$

где $P(.)$ – вероятность появления входной/выходной разности для тестируемой функции (для разностного анализа) или значения корреляции (для линейного анализа) [57].

Следующие обозначения и определения являются стандартными в линейном криптоанализе.

Определение 1. Итеративный блочный шифр – это алгоритм, который преобразует блок открытого текста фиксированного размера n в блок шифр-текста идентичного размера под воздействием ключа k путем применения итеративного обратимого преобразования p, называемого раундовым преобразованием. Обозначая открытый текст как x_0, а шифр-текст как x_R, операция зашифрования может быть записана как:

$$x_{r+1} = p_{k_r}(x_r), \quad r = 1, 2, \dots, R, \quad (39)$$

где k_r являются подключами, сгенерированными алгоритмом выработки ключей. Для простоты мы рассматриваем n-битные ключи и подключи.

Определение 2. Пусть $F: \{0,1\}^n \to \{0,1\}^n$ – биективное преобразование, a, b – две маски $\in \{0,1\}^n$. Если $X \in \{0,1\}^n$ равномерно распределенная случайная величина, тогда смещение линейной аппроксимации $LB(a, b)$ определено, как:

$$LB(a,b) = \underset{X}{Pr}\{a * X = b * F(X)\} - \frac{1}{2}, \quad (40)$$

где $*$ – скалярное произведение. Если F параметризовано ключом K, запишем $LB(a,b,K)$ и ожидаемое линейное отклонение $ELB(a,b)$ определено, как:

$$ELB(a,b) = \underset{K}{E}\big(LB(a,b,K)\big). \quad (41)$$

Линейное отклонение может быть вычислено для различных преобразований, например, для одного $sbox$, раундовой функции или блочного шифра. Точное определение линейного отклонения является вычислительно сложной задачей по мере увеличения размерности преобразования.

Определение 3. Одноraундовая характеристика для раунда i итерационного блочного шифра представляет собой пару n-битных векторов $\langle a_i, b_i \rangle$, соответствующих входным и выходным маскам для этого раунда. R-раундовая характеристика для раундов $1 \dots R$ представляет собой $(R+1)$-кортеж n-битных векторов $\Omega = \langle a_1, a_2, \dots, a_{R+1} \rangle$, где $\langle a_i, a_{i+1} \rangle$ соответствуют входным и выходным маскам для раунда i.

Определение 4. Шифр Маркова – блочный шифр, в котором линейные (и разностные) отклонения разных раундов независимы друг от друга, предполагая, что в разных раундах используются равномерно-случайные подключи [57].

В нашем случае PD-sbox-ARX является небиективным преобразованием, однако, возможным вариантом его применения является использование в качестве нелинейной функции Фейстель-подобного итерационного преобразования, представленного на рисунке 3.1. Которое, как известно, является взаимооднозначным преобразованием. Возможность применения небиективного преобразования является важным преимуществом сети Фейстеля и такие известные криптоалгоритмы как DES и ГОСТ [58, 59] используют небиективную функцию. Стоит отметить, что изначально понятия

итерационной характеристики было введено для анализа криптоалгоритма DES [60].

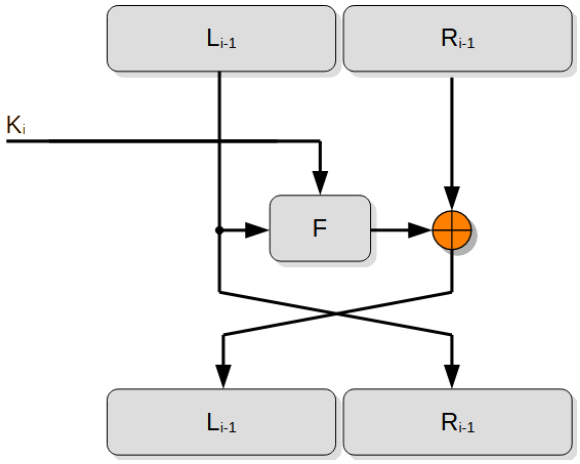


Рисунок 3.1 – Раунд сети Фейстеля

PD-sbox-ARX-32 является невзаимнооднозначной псевдо-динамической функцией, несмотря на это, её анализ с использованием SAT/SMT-решателей возможен и найденные характеристики будут валидны. В качестве примера следует привести работу [61], в которой представлены результаты первого криптоанализа шифра DES с использованием SAT-решателей.

3.1. Исследование дифференциальных и линейных характеристик PRF pCollapserARX, используя CASCADA

В рамках первоначального исследования криптографических характеристик PD-sbox-ARX, а также конструкций на их основе, реализован синтез и анализ псевдослучайной функции pCollapserARX с использованием библиотеки CASCADA.

Структура псевдослучайной функции «Collapser» (черная дыра) впервые представлена на CTCrypt’2015 [62]. Данная PRF состоит из последовательно соединённых псевдо-динамических операций подстановки. Представленная структура демонстрирует возможность применения PD-sbox в криптографических преобразованиях. В [64] предложена PRF "pCollapser" (параллельный Collapser), в которой устранен ряд недостатков "Collapser". В

"pCollapser" все блоки псевдо-динамической подстановки PD-sbox работают параллельно и независимо друг от друга в пределах одного раунда.

Высокопроизводительная PRF pCollapserARX256-32x2 представлена на конференции РусКрипто’2022 [52] – это программно-ориентированная псевдослучайная функция, основанная на PD-sbox, в которой каждая псевдо-динамическая операция подстановки состоит из 4 ARX функций (небиективных функции, использующих исключительно операции сложения по модулю, циклического сдвига и XOR).

Основное предназначение данной PRF – использование в качестве высокопроизводительной псевдослучайной функции в режимах, где не требуются обратимые раундовые преобразования: AEAD, CTR, Sponge и т.д.

Производительность PRF pCollapserARX256-32x2 с использованием компилятора Usuba [63] и набора инструкций составила: AVX256 – 2,8 такта/байт; AVX512 – 1,8 такта/байт (на одно ядро процессора Intel Core i7-11700K).

Основная идея, заложенная в PRF pCollapserARX256-32x2 – применение в качестве нелинейного элемента PD-sbox – специальной функции, позволяющей кардинально изменить свойства группы вложенных в неё функций [29–32]. Изначально, в качестве вложенных функций предполагалось применение фиксированных биективных подстановок sbox, но проведённые исследования показали, что в качестве вложенных функций хорошо подходит применение ARX-функций с изначально слабыми криптографическими свойствами.

Продемонстрировано, что объединение 4 слабых ARX-функций в PD-sbox позволяет получить свойства эквивалентных операций подстановки, близких к свойствам случайно сформированных sbox той же размерности.

3.1.1. Описание версии исследуемой псевдослучайной функции "pCollapserARX"

Рассматриваемые PRF pCollapserARX32-4x2, pCollapserARX64-8x2 и pCollapserARX128-16x2 построены по идентичному принципу:

1. Использование 4 раундов преобразования.

2. Каждый раунд содержит 16 параллельных ARX-функций.

3. Все ARX-функции собраны в четыре PD-sbox.

4. Для обеспечения динамического режима работы PD-sbox используются и формируются входные/выходные управляющие состояния.

5. Мастер ключ инициализирует внутреннее состояние, раундовые ключи не используются (аналогично потоковым шифрам).

6. Первый раунд является подготовительным, в нем PD-sbox работают в статическом режиме, но формируют управляющие значения для следующего раунда.

Обозначения в названии псевдо-случайной функции (pCollapserARX256-32x2): p – параллельный; ARX – тип используемых операций; 256 – размер блока в битах (L_{block}); 32x2 – размер слова в битах (L_{word}) (а также размерность входов/выходов ARX-функций); 32 – размер подслова в ARX-функции (использование 32 битных ARX-операций).

Используемые параметры: $N_{words} = 4$ – количество входных слов в блоке; $N_{rows} = 4$ – количество ARX-функций в одном PD-sbox; $L_{block} = L_{word} \times 4$, размер блока в битах; $L_{key} = (L_{block})$; $(2 \times L_{block})$ or $(4 \times L_{block})$, размер мастер-ключа в битах.

Используемые векторы: $m = \{m_0, m_1, m_2, m_3\}$ – вектор входного сообщения; $c = \{c_0, c_1, c_2, c_3\}$ – вектор шифр-текста; $s = \{s_{00}, s_{01}, \dots, s_{33}\}$ – вектор значения управляющего состояния; $d = \{d_{00}, d_{01}, \dots, d_{33}\}$ – раундовые константы.

В ходе исследования криптографических характеристик программой CASCADA в структуру pCollapserARX были внесены следующие изменения.

Данные изменения призваны исправить недостатки изначально предложенной версии pCollapser, в том числе приводят к увеличению весов линейных и разностных характеристик:

1. Размер внутреннего управляющего состояния увеличен до:

$$L_{control\_state} = N_{rows} \times N_{words} \times L_{word}, \text{бит.} \quad (42)$$

2. Изменена процедура формирования/обновления управляющего состояния.

3. Добавлена генерация "расширенного ключа". "Расширенный ключ" включает в себя исходное управляющее состояние и ключ первого раунда. Для раундов 2–4 нет раундовых ключей, вместо этого используется обновляемое управляющее состояние.

Параметры PRF семейства pCollapserARX представлены в таблице 3.1.

Таблица 3.1 – Параметры PRF семейства pCollapserARX

pCollapserARX:				
params	**32-4x2**	**64-8x2**	**128-16x2**	**256-32x2**
L_{block}, бит	32	64	128	256
L_{word}, бит	8	16	32	64
N_{rounds}	4	4	4	4
N_{words}	4	4	4	4
N_{rows}	4	4	4	4
L_{key}, бит	32	64	128	256
$L_{control\_state}$, бит	128	256	512	1024

3.1.2. Структура используемой ARX-функции

Структура ARX-функций выбрана исходя из обеспечения криптографических свойств (в составе PD-sbox) и обеспечения оптимального использования возможностей процессоров и аппаратных платформ.

В таблице 3.2 представлены параметры ARX-функций для PRF pCollapserARX128-16x2.

Таблица 3.2 – Параметры ARX-функций PRF pCollapserARX128-16x2

	t0	t1	t2	t3	t4	t5	t6	t7
funcARX0:	4	8	8	4	4	8	0	0
funcARX1:	4	8	4	8	8	4	4	4
funcARX2:	8	4	4	8	4	8	8	8
funcARX3:	8	4	8	4	8	4	12	12

Для получения значений параметров ARX-функций для pCollapserARX256-32x2 необходимо удвоить значения из таблицы 3.2. Аналогично получаются параметры меньших версий, только значения параметров пропорционально уменьшаются.

На рисунке 3.2 приведена структура используемых ARX-функций.

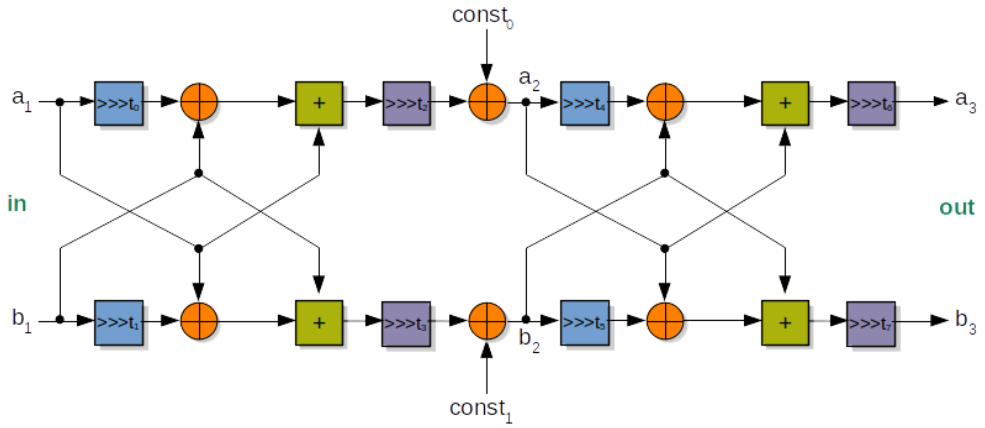


Рисунок 3.2 – Структура используемых ARX-функций

3.1.3. Функция псевдо-динамической операции подстановки

На рисунке 3.3 приведена псевдо-динамическая операция подстановки, состоящая из четырёх параллельно включенных ARX-функций. Размерность входа-выхода PD-sbox соответствует размерности используемых ARX-функций.

Выражение, описывающее значение на выходе PD-sbox:

$$c_i = \bigoplus_{j=0}^{3} funcARX_j(m_i \oplus s_j^i), \quad (43)$$

где: i – индекс n-битного слова из входного/выходного вектора и далее индекс PD-sbox; j – индекс компонента PD-sbox ; m_i – n-битные слова из входного вектора; c_i – n-битные слова из выходного вектора; funcARX – ARX-функция (компоненты PD-sbox); s_j^i – n-битные слова из входного вектора управляющего состояния (индивидуальные для каждого PD-sbox).

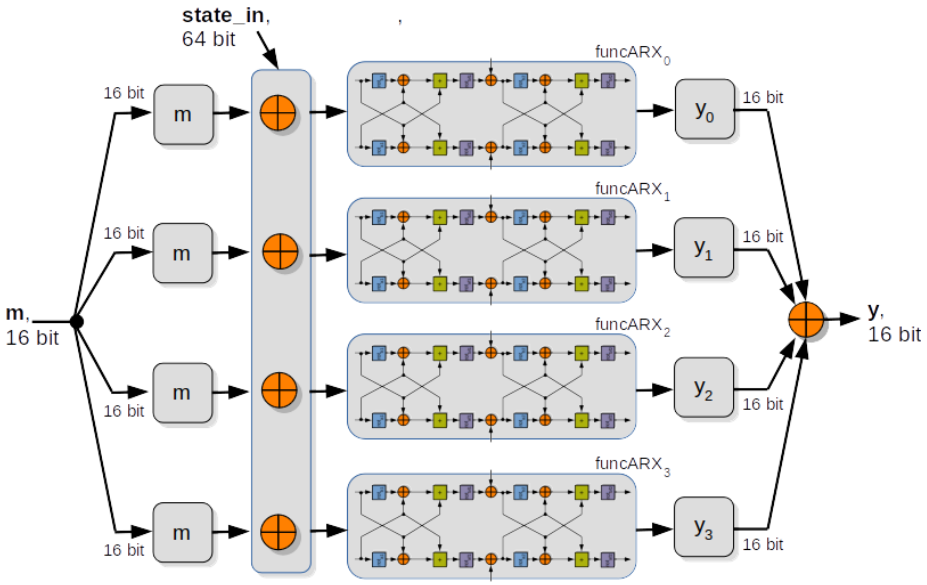


Рисунок 3.3 – Псевдо-динамическая операция подстановки для pCollapserARX64-8x2

Выражение, описывающее индивидуальные управляющие состояния на выходе PD-sbox:

$$g_n^i = c_i \oplus funcARX_j(m_i \oplus s_j^i) = \bigoplus_{n=0, n \neq i}^{3} funcARX_j(m_i \oplus s_j^i). \quad (44)$$

Фактически, с помощью этого выражения мы реализуем 4 суб-PD-sbox (по 3 ARX-функции в каждой).

Для псевдо-динамической операции подстановки PD-sbox выполнен первичный анализ линейных [29, 30] и разностных [31] свойств, предложен вычислительно эффективный метод определения усредненного распределения дифференциалов [65] и линейных свойств [37], найден класс PD-sbox,

обладающий идеальным усредненным распределением дифференциалов в статическом режиме работы [32].

В [52] показано, что объединение 4 откровенно слабых ARX-функций в PD-sbox позволяет получать свойства эквивалентных операций подстановки sbox, близкие к свойствам случайно сформированных подстановок аналогичной размерности.

3.1.4. Раунд PRF pCollapserARX

Рисунок 3.4 демонстрирует структуру одного раунда PRF pCollapserARX, где: $m = \{m_0, m_1, m_2, m_3\}$ – вектор сообщения на входе; $c = \{c_0, c_1, c_2, c_3\}$ – вектор шифр-текста; $s = \{s_{00}, s_{01}, \dots, s_{33}\}$ – вектор управляющего состояния; $d = \{d_{00}, d_{01}, \dots, d_{33}\}$ – раундовые константы, funcARX0 ... funcARX3 – ARX-функции.

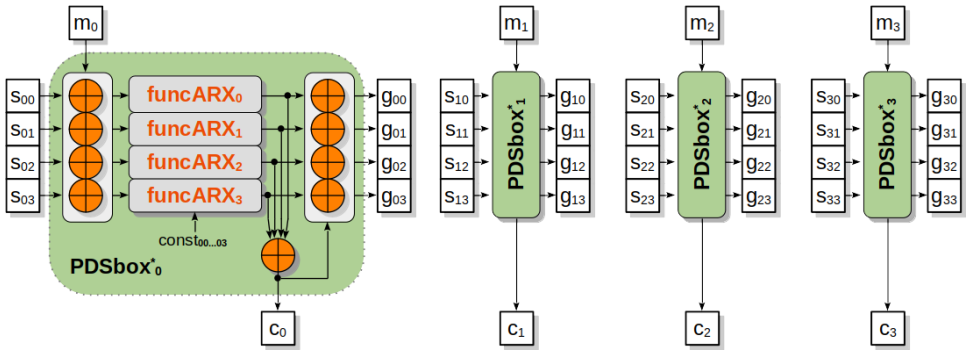


Рисунок 3.4 – Структура одного раунда PRF pCollapserARX (без новых элементов формирования управляющего состояния)

На выходе каждого раунда формируется управляющее состояние для последующих раундов. Процесс включает два шага:

1. Формирование объединённого управляющего состояния $s^* = \{s_0^*, s_1^*, s_2^*, s_3^*\}$ продемонстрировано на рисунке 3.5;

2. Распределение s^* по индивидуальным управляющим состояниям $s = \{s_{00}, s_{01}, \dots, s_{33}\}$ для каждой отдельной PD-sbox – рисунок 3.6.

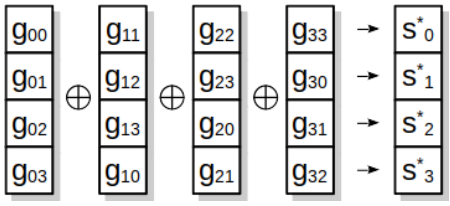


Рисунок 3.5 – Управляющее состояние на выходе. Шаг 1

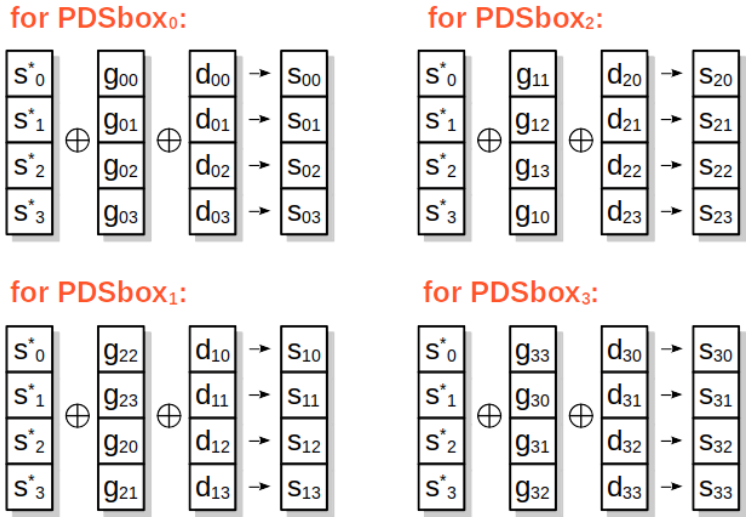


Рисунок 3.6 – Управляющее состояние на выходе. Шаг 2

Выражение, описывающее функцию, формирующую вектор нового значения управляющего состояния на выходе:

Шаг 1:

$$g^i = \left(g_0^i, g_1^i, g_2^i, g_3^i\right) = g^i <<< (i * L_{word}), \tag{45}$$

$$s^* = (s_0^*, s_1^*, s_2^*, s_3^*) = \bigoplus_{i=0}^{3} g^i. \tag{46}$$

Шаг 2:

$$s_j^i = s_j^* \oplus d_j^i \oplus g_j^i, \tag{47}$$

где $g^i = \left(g_0^i, g_1^i, g_2^i, g_3^i\right)$ – выходные значения управляющего состояния для каждого i-го PD-sbox; $a <<< b$ – циклический побитовый сдвиг вектора a на b элементов влево; d_j^i – n-битные слова вектора константы/деколлизии (индивидуально для каждого PD-sbox).

3.1.5. Генерация "расширенного ключа"

Рисунок 3.7 (левая часть) демонстрирует генерацию "расширенного ключа" для "pCollapserARX64".

Эта процедура заключается в простой загрузке мастер-ключа во внутреннее управляющее состояние и выполнении 4 раундов преобразования PRF pCollapserARX. Начальное значение входа m устанавливается равным нулю. Операция "расширенного ключа" – это инициализированное внутреннее управляющее состояние и значение выходного блока на четвертом раунде.

Для "pCollapserARX64" с размером ключа 64 бит мастер-ключ при загрузке во внутреннее управляющее состояние копируется 4 раза. Для "pCollapserARX64" с размером ключа 128 бит мастер-ключ при загрузке во внутреннее управляющее состояние копируется 2 раза.

На рисунке 3.7 (правая часть) показана структура PRF pCollapserARX при использовании 4 раундов преобразования. Входной открытый текст смешивается с частью "расширенного ключа", соответствующей выходу четвёртого раунда процедуры разворачивания ключа. Для улучшения динамического режима работы второго раунда, на его управляющий вход подмешивается входное сообщение.

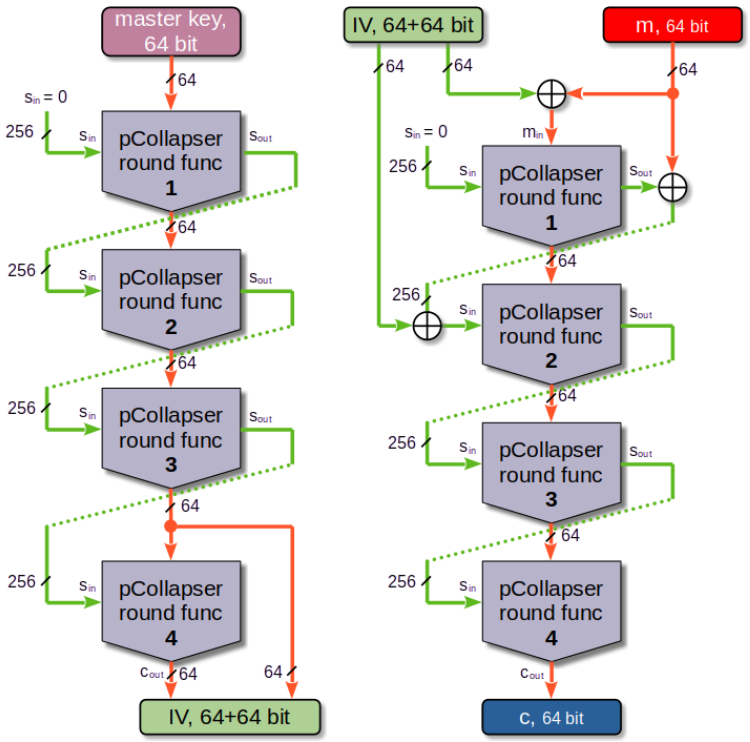


Рисунок 3.7 – Генерация «расширенного ключа» (слева) и основные раунды "pCollapserARX64" (справа)

Перед шифрованием необходимо использовать функцию генерации «расширенного ключа» для подготовки исходного внутреннего состояния "pCollapserARX64". Для каждой операции шифрования блока открытого текста используется одно и то же начальное внутреннее состояние.

3.1.6. Анализ криптографических характеристик PRF "pCollapserARX"

SAT/SMT решатели – это эффективный инструмент автоматического поиска оптимальных (для криптоанализа) характеристик криптографических функций, активно используемый в последние годы [70, 67].

Ranea A. and Rijmen V. в [43] предложили развитый инструмент CASCADA — (Characteristic Automated Search of Cryptographic Algorithms for Distinguishing Attacks) — это специализированная библиотека с открытым исходным кодом на языке программирования Python3, предназначенная для оценивания стойкости криптографических примитивов (особенно блочных

шифров) к атакам различения с использованием бит-векторных SMT-решателей. Библиотека доступна в [70].

«Инструмент CASCADA реализует структуру бит-векторных свойств и несколько методов автоматического поиска на основе SMT решателей для оценки безопасности шифров против разностных, ключезависимых разностных, rotational-XOR, невозможного разностного, невозможного rotational-XOR, ключезависимого невозможного разностного, линейного и криптоанализа с нулевой корреляцией» [43].

Для поиска характеристик при помощи CASCADA нами был создан файл "pCollapserARX_full.py", содержащий реализацию PRF pCollapserARX и файл "test_collapser.py", позволяющий запускать поиск различных криптографических характеристик для PRF pCollapserARX [45].

Для поиска характеристик использовался следующий компьютер и программное обеспечение: AMD Ryzen 5 2600, 32GB RAM, OS Ubuntu 22.04, Python version 3.10, CASCADA версии, загруженной 24 февраля 2023 года из [70].

Полученные результаты приведены в таблицах 3.3 – 3.5. В колонке "Correlation" показаны веса для линейных характеристик; в колонке XorDiff показаны значения весов для XOR-разностей, в колонке RXDiff – rotational-XOR разностей.

По умолчанию поиск характеристик ведётся по мере увеличения веса: поиск начинается с минимального веса (например, $w = 0$) и, в случае отсутствия решения от SAT-решателя, значения веса увеличивается на единицу и осуществляется повторная попытка поиска решения.

Для pCollapserARX64 и pCollapserARX128 поиск характеристик занимает значительное время (превышающее 5 дней). С увеличением значения тестируемого веса время анализа существенно увеличивается, поэтому в таблицах 3.4 и 3.5 показаны только те нижние значения, для которых SAT-решатель не смог найти решения и в явном виде это отобразил.

Необходимо отметить, что поиск решения SAT-решателем при заведомо больших значения w (относительно реального веса) происходит значительно быстрее. Таким образом, выставляя заведомо большие значения w мы смогли за разумное время в явном виде получить верхние значения весов.

Для случаев, когда поиск характеристик не осуществлялся, в таблице поставлен символ "−". Для поиска характеристик Related-Key XorDiff использовалось два раунда для формирования "расширенного ключа" (вместо 4). В последней строке каждой таблицы приведена граница применимости для соответствующего метода криптоанализа с учётом размерности входов/выходов.

Таблица 3.3 Веса криптографических характеристик pCollapserARX32

$w = -\log_2 P(.)$				
Nrounds	**Correlation**	**XorDiff**	**Related-Key XorDiff**	**RXDiff**
1	8	16	16	1276
2	26	32	64	2510 < w
3	37	32	64	3060 < w
4	50	32	64	–
bounds	16	32	32	32

Таблица 3.4 – Веса криптографических характеристик pCollapserARX64

$w = -\log_2 P(.)$			
Nrounds	**Correlation**	**XorDiff**	**RXDiff**
1	11	19	1508 < w < 1603
2	42	88	–
3	55 < w < 88	88	–
4	61 < w	88	–
bounds	32	64	64

Таблица 3.5 – Веса криптографических характеристик pCollapserARX128

$\boldsymbol{w = -\log_2 P(.)}$			
Nrounds	**Correlation**	**XorDiff**	**RXDiff**
1	11	19	1136 < w
2	49 < w < 63	110 < w < 180	–
3	–	–	–
bounds	64	128	128

В соответствии с таблицей 3.3 можно увидеть, что после 2 раунда вес (Related-Key) XorDiff характеристики не увеличивается. Анализ найденных характеристик показывает наличие "коллизий" на выходах раундов, что проявляется в "склеивании" выходных разностей на последующих раундах.

Найденные разностные характеристики для pCollapserARX32 (оказавшиеся недействительными), показывающие "склеивание" выходных разностей на 3 и 4 раундах:

$$1: Ch(w = 16, id = fa\ 00\ 00\ 00, od = 8c\ 00\ 00\ 00)$$

$$2: Ch(w = \mathbf{32}, id = \mathbf{00\ 00\ 00\ fa}, od = \mathbf{00\ 00\ 00\ 48})$$

$$3: Ch(w = \mathbf{32}, id = \mathbf{00\ 00\ 00\ af}, od = \mathbf{00\ 00\ 00\ 00})$$

$$4: Ch(w = \mathbf{32}, id = \mathbf{00\ 00\ 00\ af}, od = \mathbf{00\ 00\ 00\ 00})$$

Однако экспериментальное исследование миниверсий pCollapserARX показывает, что в реальности вероятность коллизии на выходе второго и последующих раундов соответствует вероятности коллизии на выходе случайной функции (аналогичной размерности). Благодаря наличию активно обновляемого внутреннего состояния, значительно превышающего размер входа/выхода, наличие коллизии на выходе второго раунда не приведет к склеиванию выходных значений в последующих раундах.

Несоответствие найденных и реальных характеристик объясняется тем, что разностная модель SAT-решателя построена с учётом гипотезы стохастической эквивалентности (также известной как предположение о Марковском шифре [71]). Это позволяет разбивать и анализировать

итерационные функции по частям, тем самым резко снижая сложность поиска характеристик.

Результаты исследования показывают, что гипотеза стохастической эквивалентности не справедлива для pCollapserARX (распространение разностных свойств в рамках характеристики не является статистически независимым из-за характера динамической работы PD-sbox) и отсутствие раундовых ключей (используется обновляемое внутреннее управляющее состояние). Это проявляется в виде большого количества найденных неверных (несовместимых) характеристик.

В таблицах 3.3 – 3.5 представлены веса для первых найденных характеристик без проверки их корректности, чтобы показать нижнюю границу значений весов для pCollapserARX. Реальные значения будут выше, но для определения реальных оптимальных характеристик нужны дополнительные ресурсоёмкие исследования.

Проблема затрагивает также определение характеристик для relatedkey differential, impossible-differential, related-key impossible-differential and zero-correlation cryptanalysis.

В данном случае использована экспериментальная программа-скрипт, созданная Ranea и основанная на идеях [66, 67]. Как отмечено в описании программы: «В этом скрипте мы строим модель, которая одновременно описывает переходы разности и переходы значений. Таким образом, характеристики, найденные для построенной модели, гарантированно будут действительными и мы можем восстановить из SAT-решателя все соответствующие характеристики». Стоит обратить внимание, что этот скрипт является экспериментальным. Скрипт доступен по адресу (Github: github.com/ranea/CASCADA/blob/master/
cascada/experimental/diffvalchsearch.py).

Результаты, полученные с помощью этого скрипта, представлены в таблице 3.6:

Таблица 3.6 – Веса криптографических характеристик, при использовании diffvalchsearch.py

$w = -\log_2 P(.)$				
Nrounds	**pCollapserARX16**	**pCollapserARX32**	**pCollapserARX64**	**pCollapserARX128**
1	13	18	21	19
2	48	73<w	100<w	100<w
3	75	–	–	–
bounds	16	32	64	128

Найденные разностные характеристики с использованием традиционного подхода (для мини-версии pCollapserARX16):

$$1: Ch(w = 11, id = 0\ 0\ f\ 0, od = 0\ 0\ 6\ 0)$$

$$2: Ch(w = 23, id = f\ 0\ 0\ 0, od = \mathbf{7\ 0\ 0\ 0})$$

$$3: Ch(w = 24, id = 0\ 0\ 0\ f, od = \mathbf{0\ 0\ 0\ 0})$$

$$4: Ch(w = 24, id = 0\ 0\ 0\ f, od = \mathbf{0\ 0\ 0\ 0})$$

Найденные действительные разностные характеристики с использованием экспериментального скрипта (для pCollapserARX16):

$$1: Ch(w = 13, id = 0\ d\ 0\ 0, od = 0\ 8\ 0\ 0)$$

$$2: Ch(w = 48, id = 0\ 0\ 8\ 0, od = 3\ 8\ a\ f)$$

$$3: Ch(w = 75, id = 0\ 0\ 8\ 0, od = f\ 0\ b\ 0)$$

Как можно видеть на примере мини-версии pCollapserARX16, полученные действительные характеристики не «склеиваются» (выходная разница не равна нулю), а значение веса увеличивается с каждым раундом. Данные результаты показывают, что гипотеза стохастической эквивалентности не справедлива для pCollapserARX. Проведён поиск оптимальных характеристик при помощи CASCADA для известных алгоритмов (например в AES, SPECK, SKINNY, NOEKEON, RECTANGLE) с использованием предположения о Марковском шифре, и с использованием

скрипта diffvalchsearch.py. Веса полученных характеристик совпадают, в том числе с известными опубликованными результатами.

Применение подобранных ARX-функций для использования в структуре псевдо-динамических подстановок псевдослучайной функции pCollapserARX не позволяет применять для криптоанализа широко используемые и зарекомендованные методы анализа на основе получения итерационных характеристик (гипотеза о стохастической эквивалентности), в частности библиотеку CASCADA, и требует поиска новых моделей для SAT-решателей. Необходимо, чтобы сложность синтезируемой модели была доступна для вычислений на существующих вычислительных платформах.

3.1.7. Выводы

Впервые представлено публично доступное описание PRF pCollapser для фреймворка CASCADA, позволяющий формировать SAT-модели и проводить поиск оптимальных криптографических характеристик.

Результаты поиска оптимальных криптографических характеристик при помощи CASCADA демонстрируют устойчивость первых 2 раундов (из 4) псевдослучайной функции pCollapserARX к построению различителей для таких атак, как линейный анализ, разностный анализ, Related-Key разностный анализ, RX-Differences анализ и Related-Key разностный анализ. Учитывая то, что 1-й раунд не работает в динамическом режиме, основной вклад в стойкость (вес характеристик) даёт второй раунд, работающий в динамическом режиме. Это подтверждает корректность как применения в качестве нелинейного преобразования псевдо-динамических подстановок, так и использование в их составе откровенно слабых ARX-функций.

Результаты исследований свидетельствуют о том, что применение подобранных ARX-функций для использования в структуре псевдо-динамических подстановок псевдослучайной функции pCollapserARX не позволяет применять для криптоанализа широко используемые и зарекомендованные методы анализа на основе получения итерационных

характеристик (гипотеза о стохастической эквивалентности), в частности CASCADA, и требует поиска новых моделей для SAT-решателей. Необходимо, чтобы сложность синтезируемой модели была доступна для вычислений на существующих вычислительных платформах.

Результаты исследований, представленные в главе, отражены в статьях [50, 51] в сборниках, цитируемых в РИНЦ. Результаты исследований апробированы на XII симпозиуме «Современные тенденции в криптографии» (CTCrypt 2023) 6-9 июня 2023, г. Волгоград), а также на ежегодной научно-практической конференции «РусКрипто» (РусКрипто’2022), научный доклад «Высокопроизводительная псевдослучайная функция pCollapserARX256-32x2» (Москва, 22 – 25 марта 2022 г.), на VIII Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности» (4 – 9 апреля 2022 г.) и IX Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности» (10 – 15 апреля 2023 г.).

3.2. Особенности программной реализации на малоресурсных процессорах

Реализация криптографических преобразований на микроконтроллерах (малоресурсных процессорах) является важнейшей задачей, обусловленной как повсеместным использованием криптографических преобразований в сетевых протоколах, так и значительной ресурсоёмкости этих преобразований, что может привести к дефициту вычислительных ресурсов для основных процессов.

Стоит отметить, что на различных процессорах/микроконтроллерах для ARX-операций может требоваться разное количество тактов на их выполнение. Однако, для относительно простых микроконтроллеров

предполагается, что на выполнение операций ADD или XOR требуется один такт [68]. Однако, выполнение операции ROL может требовать значительного количества инструкций и тактов микроконтроллера.

В качестве таких процессоров/микроконтроллеров рассмотрим широко распространённые 8-битные микроконтроллеры архитектуры AVR фирмы Atmel, в частности микроконтроллеры ATmega328P (применяется, например, в Arduino UNO R3) и микроконтроллеры на базе инструкций MIPS32/MIPS64 (самым известным является семейство микроконтроллеров PIC32 от Microchip).

В работе не рассмотрены более совершенные процессоры/микроконтроллеры на основе архитектур RISC-V и ARM, в которых уже реализована встроенная операция ROL, требующая небольшого количества тактов на выполнение. Что касается 8-битных микроконтроллеров семейства AVR (например, ATmega328P), то в них аппаратно реализована инструкция циклического сдвига только на 1 бит.

Для определения количества инструкций на операции циклического сдвига с разным количеством сдвигаемых бит использован ресурс godbolt.org [69], интерфейс которого представлен на рисунке 3.8, позволяющий в интерактивном режиме, как вывести результат компиляции исходного кода в виде набора ассемблерных инструкций, так и легко выбрать компилятор и архитектуру/семейство целевого процессора или микроконтроллера.

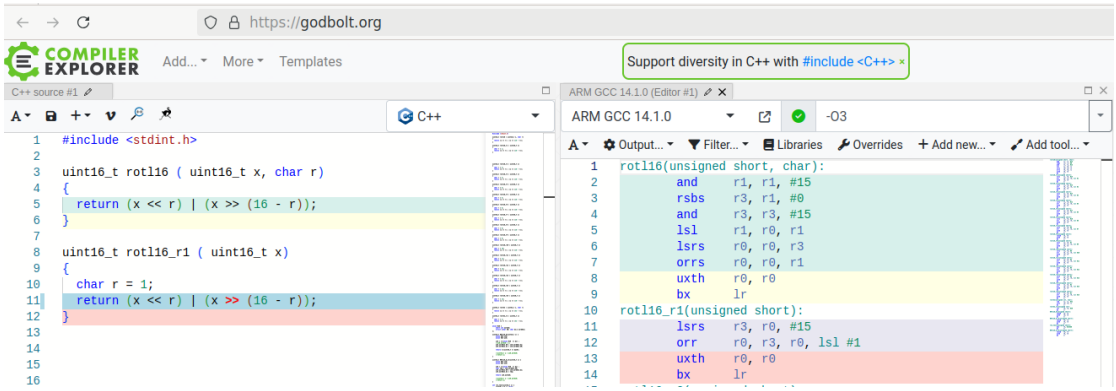


Рисунок 3.8 – Результат компиляции на ресурсе godbolt.org

Выбран типовой способ описания операции ROL на языке C в виде типовой конструкции из двух нециклических сдвигов (влево и вправо) и объединения результатов при помощи операции OR или XOR. Как известно, современные компиляторы, в том числе gcc, хорошо распознают такую типовую конструкцию и при компиляции заменяют её на соответствующую встроенную команду. Или, в случае отсутствия такой команды – на оптимальный набор инструкций, выполняющий эквивалентное преобразование.

Для каждого значения циклического сдвига реализована отдельная функция, представленная на рисунке 3.9, что позволило в дальнейшем оценить количество затрачиваемых инструкций для ROL с разным значением сдвига.

```
1  uint16_t rotl16_r1 (uint16_t x) {char r = 1; return (x << r) | (x >> (16 - r));}
2  uint16_t rotl16_r2 (uint16_t x) {char r = 2; return (x << r) | (x >> (16 - r));}
3  uint16_t rotl16_r3 (uint16_t x) {char r = 3; return (x << r) | (x >> (16 - r));}
4  ...
5  uint16_t rotl16_r15 (uint16_t x) {char r = 15; return (x << r) | (x >> (16 - r));}
```

Рисунок 3.9 – Отдельные функции для значений циклического сдвига

После компиляции для каждой функции подсчитано количество требуемых на реализацию ассемблерных инструкций. При этом инструкция RET (возврат из функции) не учтена, так как при компиляции целиком всей ARX-функции обычно осуществляется встраивание кода ROL-функции непосредственно в точку вызова функции (вместо её фактического вызова, что позволяет снизить количество инструкций вызова и возврата). В таблице 3.7 приведён пример результата компиляции, использовался флаг «-O3».

Таблица 3.7 – Пример результата компиляции 16-битовых операций ROL-1 и ROL-2 для различных целевых платформ

Операция	**Архитектура / компилятор**	
	x86-64 / gcc 14.2	AVR / gcc 14.1.0
ROL-1	rotl16_r1(unsigned short): mov eax, edi rol ax ret	rotl16_r1(unsigned int): .L__stack_usage = 0 lsl r24 rol r25 adc r24,__zero_reg__ ret

ROL-2	rotl16_r2(unsigned short):
	mov eax, edi
	rol ax, 2
	ret
		rotl16_r2(unsigned int):
		.L__stack_usage = 0
		mov r18,r24
		mov r24,r25
		mov r19,r18
		swap r19
		lsr r19
		lsr r19
		andi r19,lo8(3)
		lsl r25
		lsl r25
		swap r24
		lsr r24
		lsr r24
		andi r24,lo8(3)
		lsl r18
		lsl r18
		or r24,r18
		or r25,r19
		ret

В таблице 3.8 приведены сводные значения по количеству инструкций на реализацию операции 16-битового циклического сдвига для рассматриваемых архитектур.

Таблица 3.8 – Количество ассемблерных инструкций для реализации 16-битовой операций ROL в зависимости от значения циклического сдвига

Операция	**Количество инструкций для архитектуры/компилятора**				
	AVR (gcc 14.1.0)	Arduino Uno (1.8.9)	mips32/64 (gcc 14.1.0)	ARM (gcc 14.1.0)	x86-64 (gcc 14.2)
ROL-1	**3**	**3**	5	3	2
ROL-2	17	18			
ROL-3	17	18			
ROL-4	13	14			
ROL-5	17	18			
ROL-6	17	18			
ROL-7	13	14			
ROL-8	**3**	**3**			
ROL-9	13	14			
ROL-10	17	18			
ROL-11	17	18			
ROL-12	13	14			
ROL-13	17	18			
ROL-14	17	18			
ROL-15	**4**	**4**			

Следует обратить внимание на то, что иные операции (сложение по модулю, XOR) соответствуют одной ассемблерной инструкции и ими можно пренебречь.

3.3. Сравнение разработанного метода синтеза с методом случайного поиска параметров псевдо-динамической функции PD-sbox-ARX-32

Для оценки эффективности предложенного метода сформировано 100 000 случайных наборов параметров ARX-функций для PD-sbox-ARX-32,

для которых определено количество затрачиваемых ассемблерных инструкций и криптографические свойства – вес разностных характеристик Wd, вес линейных характеристик Wl. Данный универсальный метод применён для сравнения, так как иные методы синтеза параметров PD-sbox-ARX не представлены в открытой печати.

Ниже приведены результаты в виде гистограмм распределения по количеству затрачиваемых ассемблерных инструкций – на рисунке 3.10, в виде гистограмм распределения по весам разностных Wd и линейных Wl характеристик – на рисунке 3.11.

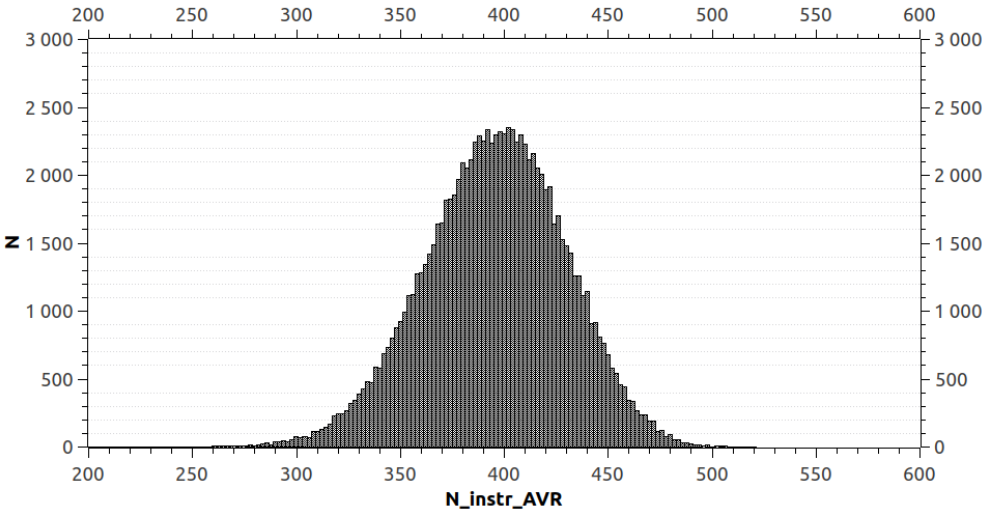


Рисунок 3.10 – Гистограмма распределения наборов параметров по количеству затрачиваемых ассемблерных инструкций

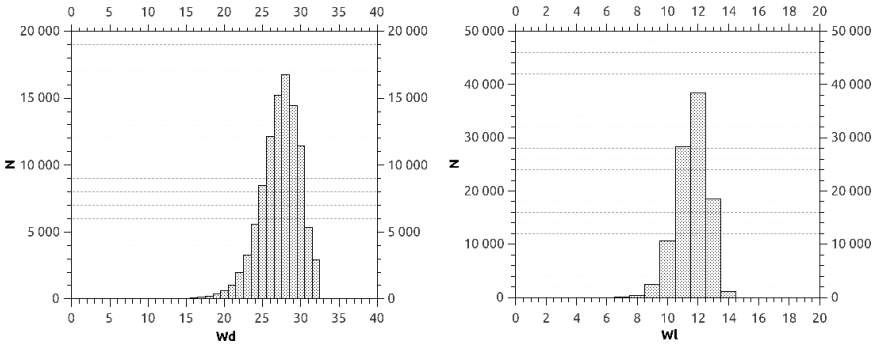


Рисунок 3.11 – Гистограмма распределения по весам разностных Wd и линейных Wl характеристик

Анализ гистограммы, представленной на рисунке 3.10, позволяет выделить 5 комбинаций параметров, выделенных на рисунке 3.12, обладающих минимальным количеством затрачиваемых ресурсов. Свойства параметров приведены в таблице 3.9.

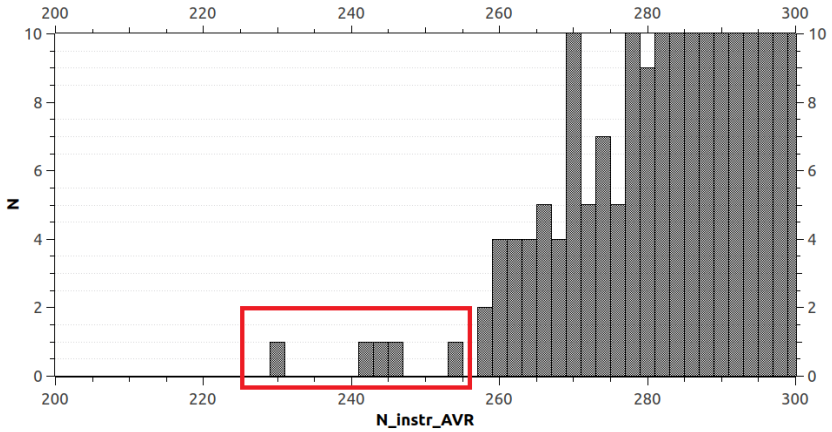


Рисунок 3.12 – Параметры, обладающие минимальным количеством затрачиваемых ресурсов при использовании метода случайного подбора

Таблица 3.9 – Свойства 5 случайно подобранных комбинаций параметров

N	**AVR**	$\boldsymbol{Wd}$	$\boldsymbol{Wl}$
1	230	24	9
2	243	20	10
3	244	18	8
4	245	20	11
5	246	25	12
Синтезированный PD-sbox-ARX	**217**	**32**	**13**

Все случайно подобранные комбинации параметров ARX-функций с минимальным количеством затрачиваемых ассемблерных инструкций обладают неприемлемыми криптографическими характеристиками. Однако, даже в данном случае они существенно уступают синтезированному PD-sbox-ARX по количеству затрачиваемых ресурсов. При сравнении с вариантом 1 из таблицы 3.9, разница составляет ~5%, учитывая его неудовлетворительные криптографические характеристики.

На рисунке 3.13 приведены результаты в виде гистограмм распределения по количеству затрачиваемых ассемблерных инструкций, минимальное значение затрачиваемых ассемблерных инструкций равно 284 при $Wd > 29$ и $Wl > 10$.

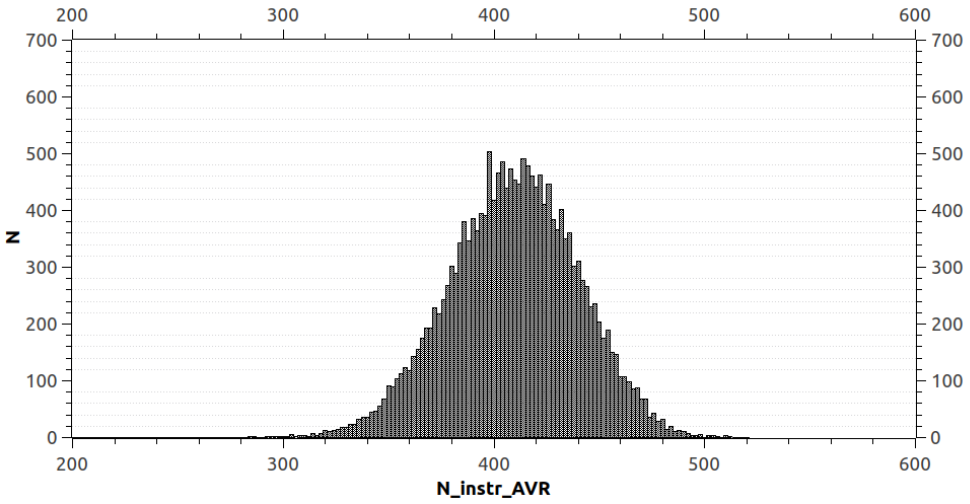


Рисунок 3.13 – Распределение наборов параметров по количеству затрачиваемых ассемблерных инструкций при $Wd > 29$ и $Wl > 10$

Анализ гистограммы, представленной на рисунке 3.13, позволяет выделить 5 комбинаций параметров, выделенных на рисунке 3.14, обладающих минимальным количеством затрачиваемых ресурсов при $Wd > 29$ и $Wl > 10$. Свойства параметров приведены в таблице 3.10.

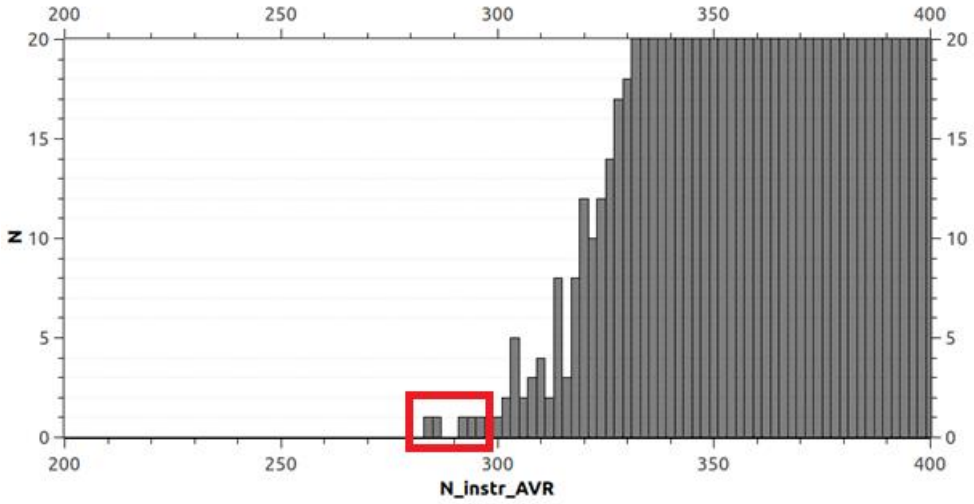


Рисунок 3.14 – Параметры, с минимальным количеством затрачиваемых ресурсов, метод случайного подбора при $Wd > 29$ и $Wl > 10$

Таблица 3.10 – Свойства 5 случайно подобранных комбинаций параметров при $Wd > 29$ и $Wl > 10$

N	**AVR**	$\boldsymbol{Wd}$	$\boldsymbol{Wl}$
1	284	30	11
2	286	30	11
3	292	30	11
4	294	31	11
5	297	30	11
Синтезированный PD-sbox-ARX	**217**	**32**	**13**

Все случайно подобранные комбинации параметров ARX-функций с минимальным количеством затрачиваемых ассемблерных инструкций обладают удовлетворимыми криптографическими свойствами, однако существенно уступают синтезированному PD-sbox-ARX по количеству затрачиваемых ресурсов. При сравнении с вариантом 1 из таблицы 3.10, разница составляет ~24%, без учёта уступающих криптографических свойств.

В свою очередь, применение разработанного метода для ранее рассмотренных криптографических преобразований – Speck32 и miniAlzette32 демонстрирует его универсальность.

Для проверки эффективности подбора параметров miniAlzette32 использовались 4 попытки (по 100 итераций подбора параметров в каждой попытке) и 8 итераций преобразования miniAlzette32. В таком случае, исходное значение параметров (значений циклических сдвигов) miniAlzette32 представляется в виде [15,12,9,9,0,15,12,8, 15,12,9,9,0,15,12,8,], а количество ассемблерных инструкций на исходную реализацию циклических сдвигов для микроконтроллеров архитектуры AVR составляет 126. Результаты поиска наилучших параметров для miniAlzette32 приведены в таблице 3.11. В одной из попыток удалось уменьшить количество затрачиваемых ресурсов на 31,7 % (со 126 до 86) для архитектуры AVR при сопоставимых значениях $Wd \geq 27$ и $Wl \geq 13$.

Необходимо отметить, что при синтезе подстановок учитываются криптографические свойства как самого sbox, так и результирующего преобразования, в котором подстановки разных итераций связаны через перемешивающие операции. Для Alzette приведены, в качестве основных, линейные и разностные свойства [46]. При этом в разработанный метод можно добавить как учёт иных криптографических свойств, так и оценку свойств результирующих преобразований. Ограничением будут выступать возможности SAT-решателя и вычислительной техники по определению криптографических свойств за приемлемое время.

Таблица 3.11 – Параметры miniAlzette32 при применении разработанного метода синтеза параметров PD-sbox-ARX, при $Wd \geq 27$ и $Wl \geq 13$

ROT	**AVR**	**Улучшение, %**
[15, 12, 9, 9, 0, 15, 12, 8] оригинальные параметры	126	–
[15, 12, 9, 9, 0, 1, 12, 8] синтезированные параметры	124	1,6
[1, 12, 15, 9, 0, 1, 12, 8] синтезированные параметры	104	17,4
[15, 12, 1, 1, 0, 15, 12, 8] синтезированные параметры	**86**	**31,7**

Для проверки эффективности подбора параметров Speck32 использовались 4 попытки (по 100 итераций подбора параметров в каждой попытке) и 8 итераций преобразования Speck32. В таком случае, исходное значение параметров (значений циклических сдвигов) Speck32 представляется в виде [7,2,7,2,7,2,7,2, 7,2,7,2,7,2,7,2] и количество ассемблерных инструкций на исходную реализацию циклических сдвигов для микроконтроллеров архитектуры AVR составляет 240. Результаты поиска наилучших параметров для Speck32 приведены в таблице 3.12. Помимо улучшения значений $Wd \geq 24$ и $Wl \geq 11$ в одной из попыток удалось уменьшить количество затрачиваемых ресурсов на 37,5 % (с 240 до 150) для архитектуры AVR.

Таблица 3.12 – Параметры Speck32 при применении разработанного метода синтеза параметров PD-sbox-ARX, при $Wd \geq 24$ и $Wl \geq 11$

ROT	**AVR**	**Улучшение, %**
[7,2,7,2,7,2,7,2] оригинальные параметры	240	–
[7, 9, 8, 2, 15, 4, 12, 2] синтезированные параметры	186	22,5
[7, 2, 7, 4, 7, 8, 1, 3] синтезированные параметры	184	23,3
[15, 3, 15, 12, 7, 14, 15, 8] синтезированные параметры	**150**	**37,5**

3.4. Выводы

Подобранная структура 32-битной ARX-функции в составе PD-sbox позволяет обеспечить критический путь (максимальное количество последовательных операций сложения по модулю 2^{16}) в четыре раза меньше, чем 8-итерационная 32-битная Alzette-подобная структура, при двухкратном увеличении количества операций и при сопоставимых максимальных значениях весов разностных и линейных характеристик.

Аналогичный результат получается при сравнении 32-битной ARX-функции с 8-итерационным 32-битным преобразованием из блочного криптоалгоритма Speck32.

Практическая ценность: при аппаратной реализации ARX-функции данное свойство позволяет пропорционально уменьшить (до 4 раз) задержку при преобразовании блоков информации.

Предложенный универсальный метод синтеза параметров 32-битной ARX-функции (для микроконтроллеров семейства AVR) позволяет получить параметры операций циклического сдвига, при которых обеспечивается максимальный вес разностной характеристики равный 2^{-32} (эмпирический вес 2^{-26}) и вес линейной характеристики 2^{-13} для результирующего PD-sbox-ARX, включающей в свой состав четыре 32-битные ARX-функции. Сопоставимые разностные и линейные характеристики имеют 8-итерационные 32-битная Alzette-подобная структура и 8-итерационное 32-битное преобразование из блочного криптоалгоритма Speck32.

Предложенный универсальный метод синтеза параметров ARX-функции, в частности размерностью 32 бит, позволяет минимизировать количество затрачиваемых ассемблерных инструкций на операции циклического сдвига при реализации на малоресурсных 8-битных микроконтроллерах семейства AVR (например, ATmega328P).

Например, при параметрах циклических сдвигов [5, 8, 8, 8, 8, 11, 0, 0], [12, 3, 8, 8, 8, 15, 4, 4], [8, 12, 8, 8, 9, 8, 8, 8], [1, 8, 8, 8, 8, 3, 12, 12] на их реализацию потребуется 217 ассемблерных инструкций микроконтроллеров архитектуры AVR. Что ориентировочно на 10% меньше, чем для типовых получаемых параметров и соответствует количеству ресурсов 8-итерационного преобразования из криптоалгоритма Speck32 (240 ассемблерных инструкций микроконтроллеров архитектуры AVR) на циклические сдвиги.

Результаты исследований, представленные в главе, опубликованы в статьях [53, 64], цитируемых в ведущих рецензируемых научных журналах, входящих в перечень ВАК РФ.

ЗАКЛЮЧЕНИЕ

В результате диссертационного исследования решена актуальная научная задача и достигнута поставленная цель, заключающаяся в минимизации затрачиваемых ресурсов программной реализации криптографических преобразований при обеспечении заданных криптографических свойств посредством разработки метода синтеза псевдо-динамических подстановок на основе ARX-функций. Это подтверждается следующими полученными научными и практическими результатами:

1. Предложенный метод синтеза псевдо-динамических подстановок на основе ARX-функций позволяет получать преобразования, удовлетворяющие требованиям по криптографическим свойствам, затрачиваемым ресурсам и скорости программной реализации криптографических преобразований.

2. Разработана структура 32-битной ARX-функции, позволяющая в составе PD-sbox обеспечить критический путь (максимальное количество последовательных операций сложения по модулю 2^{16}) в четыре раза меньше, чем ARX-преобразования, такие как 8-итерационная 32-битная Alzette-подобная структура, или 8-итерационное 32-битное преобразование криптоалгоритма Speck32, при двукратном увеличении количества операций и при сопоставимых максимальных значениях весов разностных и линейных характеристик.

3. Разработан универсальный метод синтеза PD-sbox-ARX, позволяющий путём подбора параметров ARX-функций минимизировать количество затрачиваемых ассемблерных инструкций на операции циклического сдвига при их реализации на малоресурсных 8-битных микроконтроллерах архитектуры AVR. Доказательство эффективности предложенного метода приведено для микроконтроллера архитектуры AVR – ATmega328P и конкретной реализации псевдо-динамической функции PD-sbox-ARX-32. В отличие от метода случайного поиска оптимальных параметров, разработанный метод позволяет снизить количество

соответствующих ассемблерных инструкций на 23,6% при программной реализации псевдо-динамической подстановки, включающей в свой состав четыре 32-битные ARX-функции.

4. Метод синтеза псевдо-динамических подстановок на основе ARX-функций (PD-sbox-ARX) позволяет подобрать параметры для 32-битных ARX-функций, при которых, в отличие от 8-итерационного 32-битного преобразования криптоалгоритма Speck32, требуется на 10,6% меньше ассемблерных инструкций на операции циклического сдвига при их реализации на малоресурсных 8-битных микроконтроллерах семейства AVR, в частности ATmega328P, и обеспечивается максимальный вес разностной характеристики, равный 2^{-32} (эмпирический вес 2^{-26}), и вес линейной характеристики 2^{-13}.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Biryukov, A., Perrin, L. State of the Art in Lightweight Symmetric Cryptography [Текст] / Biryukov, A., Perrin, L. // IACR Cryptol. ePrint Arch.. – 2017. – С. 511.

2. Жуков, А. Е. Легковесная криптография. Часть 1 / А. Е. Жуков // Вопросы кибербезопасности. – 2015. – № 1(9). – С. 26-43.

3. Жуков, А. Е. Легковесная криптография. Часть 2 / А. Е. Жуков // Вопросы кибербезопасности. – 2015. – № 2(10). – С. 2-10.

4. ГОСТ Р 34.12–2015. Криптографическая защита информации. Блочные шифры : национальный стандарт Российской Федерации : издание официальное : утверждён и введён в действие Приказом Федерального агентства по техническому регулированию и метрологии от 19 июня 2015 г. № 749-ст : введён впервые : дата введения 2016–01–01 / разработан Центром защиты информации и специальной связи ФСБ России с участием Открытого акционерного общества «Информационные технологии и коммуникационные системы» (ОАО «ИнфоТеКС»). – Москва: Стандартинформ, 2016. – Текст : непосредственный.

5. PRESENT: An ultra-lightweight block cipher / A. Bogdanov, G. Leander, C. Paar [et al.] // Lecture Notes in Computer Science. – 2007. – Vol. 4727 LNCS. – P. 450-466. – DOI 10.1007/978-3-540-74735-2_31.

6. Shirai, T. The 128-Bit Blockcipher CLEFIA (Extended Abstract) / T. Shirai, K. Shibutani, T. Akishita. – Текст : непосредственный // Fast Software Encryption. FSE 2007. – Berlin, Heidelberg : , 2007.

7. Hong, D. A 128-Bit Block Cipher for Fast Encryption on Common Processors / D. Hong, J. K. Lee, D. C. Kim. – Текст : непосредственный // Information Security Applications. WISA 2013. – Cham, 2014.

8. NIST Issues First Call for ‘Lightweight Cryptography’ to Protect Small Electronics. – Текст : электронный // National Institute of Standards and Technology : [сайт]. – URL: https://www.nist.gov/news-events/news/2018/04/nist-

issues-first-call-lightweight-cryptography-protect-small-electronics (дата обращения: 26.01.2025).

9. Ivanov, G. Reversed genetic algorithms for generation of bijective s-boxes with good cryptographic properties / G. Ivanov, N. Nikolov, S. Nikova // Cryptography and Communications. – 2016. – Vol. 8, No. 2. – P. 247-276. – DOI 10.1007/s12095-015-0170-5.

10. Прудников, В. А. Анализ существующих подходов к синтезу псевдо-динамических sbox / В. А. Прудников // Вопросы кибербезопасности. – 2024. – № 4(62). – С. 57-64. – DOI 10.21681/2311-3456-2024-4-57-64.

11. Tesař, P. A new method for generating high non-linearity s-boxes / P. Tesař. – Текст : непосредственный // Radioengineering. – 2010. – № 19. – С. 23-26.

12. Kazymyrov, O. A method for generation of high-nonlinear s-boxes based on gradient descent / O. Kazymyrov, V. Kazymyrova, R. Oliynykov. – Текст : непосредственный // IACR Cryptology ePrint Archive (2013). – 2013. – № 578. – С. 71-78.

13. Соколов, А. В. Методы синтеза четверичных последовательностей де БрЕйна для задач криптографии / А. В. Соколов, М. И. Мазурков // Решетневские чтения. – 2012. – Т. 2. – С. 682-683.

14. Sankaralingam, A. HPAC-sbox a novel implementation of predictive learning classifier and adaptive chaotic s-box for counterfeiting sidechannel attacks in an IOT networks / A. Sankaralingam, U. Vivek. — Текст : непосредственный // Microprocessors and Microsystems. – 2021. – № 81(6). – DOI 10.1016/j.micpro.2020.103737.

15. Artuğer, F. Comparison of Nonlinearity Value of Substitution Box Generation Approaches / F. Artuğer, S. Karakuş, F. Özkaynak // International Conference on Recent Academic Studies. – 2023. – Vol. 1. – P. 46-49. – DOI 10.59287/icras.670.

16. Kökçam, A. H. A new approach to design S-box generation algorithm based on genetic algorithm / A. H. Kökçam, Ü. Çavuşoğlu // International Journal

of Bio-Inspired Computation. – 2021. – Vol. 17, No. 1. – P. 52. – DOI 10.1504/ijbic.2021.10035835.

17. S-box generation algorithm based on hyperchaotic system and its application in image encryption / S. Yang, X. Tong, Zh. Wang, M. Zhang // Multimedia Tools and Applications. – 2023. – Vol. 82, No. 17. – P. 25559-25583. – DOI 10.1007/s11042-023-14394-1.

18. Bernstein, D.J. The Salsa20 Family of Stream Ciphers [Текст] / D.J. Bernstein, M. Robshaw, O. Billet, // New Stream Cipher Designs. Lecture Notes in Computer Science. – 2008. – № 4986. – DOI 10.1007/978-3-540-68351-3_8.

19. Maitra, S. Salsa20 Cryptanalysis: New Moves and Revisiting Old Styles / S. Maitra, G. Paul, W. Meier. – Текст : непосредственный // IACR Cryptol. ePrint Arch.. – 2015. – № 217.

20. Coutinho, M. Improved Linear Approximations to ARX Ciphers and Attacks Against ChaCha / M. Coutinho, T. Neto. – Текст : непосредственный // Advances in Cryptology – EUROCRYPT 2021. – Cham, 2021. – С. 711-740. – DOI 10.1007/978-3-030-77870-5_25.

21. Improved Differential-Linear Attacks with Applications to ARX Ciphers / Ch. Beierle, M. Broll, F. Canale [et al.] // Journal of Cryptology. – 2022. – Vol. 35, No. 4. – P. 1-61. – DOI 10.1007/s00145-022-09437-z.

22. Beaulieu, R. SIMON and SPECK: Block Ciphers for the Internet of Things. / R. Beaulieu, D. Shors, J. Smith [и др.]. – Текст : непосредственный // IACR Cryptol. ePrint Arch. 2015. – 2015. – № 585.

23. Abed, F. Cryptanalysis of the Speck Family of Block Ciphers / F. Abed, E. List, S. Lucks, J. Wenzel. – Текст : непосредственный // Cryptology ePrint Archive. – 2013. – № 568.

24. WARX: efficient white-box block cipher based on ARX primitives and random MDS matrix / Ju. Liu, V. Rijmen, Yu. Hu [et al.] // Science China Information Sciences. – 2022. – Vol. 65, No. 3. – P. 132302. – DOI 10.1007/s11432-020-3105-1.

25. Beierle, C. Alzette: a 64-bit ARX-box (feat. CRAX and TRAX) / C. Beierle, D. Micciancio, T. Ristenpart. – Текст : непосредственный // Advances in Cryptology – CRYPTO 2020. – Cham, 2020. – С. 419-448. – DOI 10.1007/978-3-030-56877-1_15.

26. Weerasinghe, T. D. An Effective RC4 Stream Cipher / T. D. Weerasinghe. – Текст : непосредственный // IEEE 8th International Conference on Industrial and Information Systems. – 2013. – С. 69-74. – DOI 10.1109/ICIInfS.2013.6731957.

27. Klein, A. Attacks on the RC4 stream cipher / A. Klein. – Текст : непосредственный // Designs, codes and cryptography. – 2008. – № 48(3). – С. 269-286. – DOI: 10.1007/s10623-008-9206-6.

28. Румянцев, К. Е. Псевдо-динамические таблицы подстановки: основа современных симметричных криптоалгоритмов / К. Е. Румянцев, С. В. Поликарпов, А. А. Кожевников // Научное обозрение. – 2014. – № 12-1. – С. 162-166.

29. Поликарпов, С. В. Исследование линейных характеристик псевдодинамических подстановок / С. В. Поликарпов, К. Е. Румянцев, А. А. Кожевников // Известия ЮФУ. Технические науки. – 2015. – № 5(166). – С. 111-123.

30. Поликарпов, С. В. Псевдо-динамические подстановки: исследование линейных свойств / С. В. Поликарпов, А. А. Кожевников // Известия ЮФУ. Технические науки. – 2015. – № 8(169). – С. 19-32.

31. Поликарпов, С. В. Псевдо-динамические таблицы подстановки: исследование дифференциальных характеристик / С. В. Поликарпов, К. Е. Румянцев, А. А. Кожевников // Физико-математические методы и информационные технологии в естествознании, технике и гуманитарных науках : сборник материалов международного научного е-симпозиума, Москва, 27–28 декабря 2014 года / под редакцией А.Б. Чебоксарова. – Москва: Международный центр научно-исследовательских проектов, 2015. – С. 77-89.

32. Polikarpov, S. On a class pseudo-dynamic substitutions PD-Sbox, with a perfect averaged distribution of differentials in static mode of work / S. Polikarpov, D. Petrov, A. Kozhevnikov // ACM International Conference Proceeding Series : Proceedings of 2017 International Conference on Cryptography, Security and Privacy, ICCSP 2017, Wuhan, 17–19 марта 2017 года. – Wuhan: Association for Computing Machinery, 2017. – P. 17-21. – DOI 10.1145/3058060.3058087.

33. Прудников, В. А. Исследование нелинейных свойств псевдодинамической подстановки Pd-sbox 6x4x4 / В. А. Прудников // Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности : Сборник статей V Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов, Таганрог, 01–07 апреля 2019 года. – Таганрог: Южный федеральный университет, 2019. – С. 96-99.

34. Поликарпов, С. В. Исследование свойств миниверсии псевдо-случайной функции pcollapser / С. В. Поликарпов, В. А. Прудников, К. Е. Румянцев // Известия ЮФУ. Технические науки. – 2022. – № 6(230). – С. 148-162. – DOI 10.18522/2311-3103-2022-6-148-162.

35. Молдовян, Н. А. Криптография : от примитивов к синтезу алгоритмов: : науч. изд. / Н. А. Молдовян, А. А. Молдовян, М. А. Еремеев ; Н. А. Молдовян, А. А. Молдовян, М. А. Еремеев. – СПб. : БХВ-Петербург, 2004. – 446 с. – ISBN 5-94157-524-6.

36. Прудников, В. А. Исследование распределения нелинейных свойств эквивалентных подстановок для псевдодинамических подстановок Pd-sbox-2x8 / В. А. Прудников // Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности : Сборник статей Всероссийской научно-технической конференции, Таганрог, 06–12 апреля 2020 года. – Таганрог: Южный федеральный университет, 2020. – С. 123-127.

37. Поликарпов, С. В. Вычислительно эффективный метод определения усреднённых линейных свойств псевдо-динамических подстановок / С. В. Поликарпов, В. А. Прудников, К. Е. Румянцев // Известия

ЮФУ. Технические науки. – 2020. – № 5(215). – С. 16-30. – DOI 10.18522/2311-3103-2020-5-16-30.

38. Рябинин, И. А. Логико-вероятностный анализ проблем и надежности, живучести и безопасности: очерки разных лет / И. А. Рябинин. – Новочеркасск : Южно-Российский гос. технический ун-т (Новочеркасский политехнический ин-т), 2009. – 599 с. – Текст : непосредственный.

39. Biham, E. Differential cryptanalysis of DES-like cryptosystems / E. Biham, A. Shamir // Journal of Cryptology. – 1991. – Vol. 4, No. 1. – P. 3-72. – DOI 10.1007/bf00630563.

40. Поликарпов, С. В. Программные инструменты анализа нелинейных свойств криптографических подстановок / С. В. Поликарпов, В. А. Прудников, К. Е. Румянцев // Digital Era : Материалы I Всероссийской научно-практической конференции, Грозный, 26 марта 2021 года / Хасухаджиев А.С.-А.. – Грозный: Чеченский государственный университет, 2021. – С. 138-141. – DOI 10.36684/38-2021-1-138-141.

41. Прудников, В. А. Программный инструмент анализа нелинейных характеристик криптографических подстановок, использующий многопоточные вычисления / В. А. Прудников // Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности : сборник статей VII Всероссийской научно-технической конференции, Таганрог, 05–11 апреля 2021 года / Министерство науки и высшего образования Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования "Южный федеральный университет". – Таганрог: Южный федеральный университет, 2021. – С. 123-125.

42. Свидетельство о государственной регистрации программы для ЭВМ № 2024610931 Российская Федерация. Программа анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций : № 2023688912 : заявл. 21.12.2023 : опубл. 16.01.2024 / С. В. Поликарпов, В. А. Прудников, К. Е. Румянцев ; заявитель

федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет».

43. Ranea, A. Characteristic automated search of cryptographic algorithms for distinguishing attacks (CASCADA) / A. Ranea, V. Rijmen // IET Information Security. – 2022. – Vol. 16, No. 6. – P. 470-481. – DOI 10.1049/ise2.12077.

44. McKay, K. Analysis of ARX Functions: Pseudo-linear Methods for Approximation, Differentials, and Evaluating Diffusion / K. McKay, P. Vora. – Текст : непосредственный // IACR Cryptol. ePrint Arch. – 2014.

45. Polikarpov, S. CTCrypt2023 / S. Polikarpov, V. Prudnikov, K. Rumyantsev. – Текст : электронный // Github : [сайт]. – URL: https://github.com/pruvad/CTCrypt2023 (дата обращения: 26.01.2025).

46. Beierle, C. Alzette: A 64-Bit ARX-box / C. Beierle, D. Micciancio, T. Ristenpart. – Текст : непосредственный // Advances in Cryptology – CRYPTO 2020. – Cham, 2020. – DOI 10.1007/978-3-030-56877-1_15.

47. Triathlon of lightweight block ciphers for the Internet of things / D. Dinu, Ya. Le. Corre, D. Khovratovich [et al.] // Journal of Cryptographic Engineering. – 2019. – Vol. 9, No. 3. – P. 283-302. – DOI 10.1007/s13389-018-0193-x.

48. Beierle, C. Lightweight AEAD and Hashing using the Sparkle Permutation Family / C. Beierle, A. Biryukov, S. Cardoso, J. Großschädl, L. Perrin, A. Udovenko, V. Velichkov, Q. Wang. – Текст : непосредственный // IACR Transactions on Symmetric Cryptology. – 2020. – С. 208-261. – DOI 10.13154/tosc.v2020.iS1.208-261.

49. Beyne, T. A Geometric Approach to Linear Cryptanalysis / T. Beyne, M. Tibouchi, H. Wang. – Текст : непосредственный // Advances in Cryptology – ASIACRYPT 2021. – Cham, 2021. – С. 36-66. – DOI 10.1007/978-3-030-92062-3_2.

50. Прудников, В. А. Анализ линейных свойств псевдо-динамической подстановки на базе ARX-конструкций / В. А. Прудников // Неделя науки 2022 : Сборник тезисов : в двух частях, Ростов-на-Дону, 18 апреля – 27 2022 года /

Редакционная коллегия: Я. А. Асланов, О.В. Батычко, М. А. Лачугина, Н. П. Сохиева ; МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ, Федеральное государственное автономное образовательное учреждение высшего образования «ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ». Том Часть 1. – Ростов-на-Дону: Издательство Южного федерального университета, 2022. – С. 619-623.

51. Прудников, В. А. Анализ количества активных ARX-функций для мини-версии PRF pCollapser-ARX / В. А. Прудников // Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности : Сборник статей Всероссийской научно-технической конференции, Таганрог, 10–15 апреля 2023 года. – Таганрог: Южный федеральный университет, 2023. – С. 61-64.

52. Поликарпов, С.В. Высокопроизводительная псевдослучайная функция pCollapserARX256-32x2 / Поликарпов С.В., Румянцев К.Е., Прудников В.А. [Электронный ресурс] // Конференция РусКрипто : [сайт]. — URL: https://www.ruscrypto.ru/resource/archive/rc2022/files/02_polikarpov_rumyantsev_prudnikov.pdf (дата обращения: 26.01.2025).

53. Поликарпов, С. В. Синтез псевдо-динамических функций PD-sbox-ARX-32 / С. В. Поликарпов, В. А. Прудников, К. Е. Румянцев // Известия ЮФУ. Технические науки. – 2024. – № 5(241). – С. 102-118. – DOI 10.18522/2311-3103-2024-5-102-118.

54. Stachowiak, S. New results in SAT – cryptanalysis of the AES / S. Stachowiak, M. Kurkowski, A. Soboń. – Текст : непосредственный // IEEE 16th International Scientific Conference on Informatics (Informatics). – 2022. – С. 280-286.

55. Bellini, E. Differential Cryptanalysis with SAT, SMT, MILP, and CP: A Detailed Comparison for Bit-Oriented Primitives / E. Bellini, A.D. Piccoli, M. Formenti, D. Gérault, P. Huynh, S. Pelizzola, S. Polese, A. Visconti – Текст :

непосредственный // Cryptology and Network Security. – 2023. – С. 268-292. – DOI 10.1007/978-981-99-7563-1_13.

56. Shi, J. SAT-Based Security Evaluation for WARP against Linear Cryptanalysis / J. Shi, G. Liu, C. Li. – Текст : непосредственный // IET Information Security. – 2023. – DOI 10.1049/2023/5323380.

57. Collard, B. Experimenting linear cryptanalysis / B. Collard, F. Standaert. – Текст : непосредственный // Cryptology and Information Security Series. – 2011. – № 7. – DOI 10.3233/978-1-60750-844-1-1.

58. Diffie, W. Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard / W. Diffie, M. E. Hellman. – Текст : непосредственный // Computer. – 1997. – № 10(6). – С. 74-84. – DOI 10.1109/C-M.1977.217750.

59. ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования : государственный стандарт Союза ССР : издание официальное : утверждён и введён в действие Постановлением Государственного комитета СССР по стандартам от 02.06.89 № 1409 : введён впервые : дата введения 1990-07-01. – М. : Стандартинформ, 1990. – Текст: непосредственный.

60. Matsui, M. Linear Cryptoanalysis Method for DES Cipher / M. Matsui. – Текст : непосредственный // Advances in Cryptology – EUROCRYPT ’93, Workshop on the Theory and Application of of Cryptographic Techniques. – Lofthus, 1993. – С. 386-397. – DOI 10.1007/3-540-48285-7_33.

61. Massacci, F. Logical cryptanalysis as a SAT-problem: Encoding and analysis / F. Massacci, L. Marraro. – Текст : непосредственный // Journal of Automated Reasoning. – 2000. – № 24. – С. 165-203. – DOI 10.1023/A:1006326723002.

62. Kozhevnikov, A. A. On differential properties of a symmetric cryptoalgorithm based on pseudo-dynamic substitutions / A. A. Kozhevnikov, S. V. Polikarpov, K. E. Rumyantsev // Математические вопросы криптографии. – 2016. – Vol. 7, No. 2. – P. 91-102. – DOI 10.4213/mvk186.

63. Mercadier, D. Usuba: high-throughput and constant-time ciphers, by construction / D. Mercadier, P. Dagand. – Текст : непосредственный // 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2019). Association for Computing Machinery. – New York : , 2019. – С. 157-173. – DOI 10.1145/3314221.3314636.

64. Псевдослучайная функция PCOLLAPSER, обеспечивающая экстремальный параллелизм обработки информации / С. В. Поликарпов, В. А. Прудников, А. А. Кожевников, К. Е. Румянцев // Известия ЮФУ. Технические науки. – 2019. – № 5(207). – С. 88-100. – DOI 10.23683/2311-3103-2019-5-88-100.

65. Polikarpov, S. Computationally efficient method for determining averaged distribution of differentials for pseudo-dynamic substitutions / S. Polikarpov, K. Rumyantsev, D. Petrov // AIP Conference Proceedings : International Conference on Electrical, Electronics, Materials and Applied Science, Secunderabad, Telangana, 22–23 декабря 2017 года. Vol. 1952. – Secunderabad, Telangana: American Institute of Physics Inc., 2018. – P. 020091. – DOI 10.1063/1.5032053.

66. Sadeghi, S. Proposing an MILP-based method for the experimental verification of difference-based trails: application to SPECK, SIMECK / S. Sadeghi, V. Rijmen, N. Bagheri // Designs, Codes and Cryptography. – 2021. – Vol. 89, No. 9. – P. 2113-2155. – DOI 10.1007/s10623-021-00904-5.

67. Improved rotational-XOR cryptanalysis of Simon-like block ciphers / J. Lu, Yu. Liu, T. Ashur [et al.] // IET Information Security. – 2022. – Vol. 16, No. 4. – P. 282-300. – DOI 10.1049/ise2.12061.

68. 8-bit Atmel Microcontroller with 128Kbytes In-System Programmable Flash. – Текст : электронный // ATmega128, ATmega128L. Rev. 2467X–AVR–06/11. 2011 Atmel Corporation : [сайт]. – URL: http://ww1.microchip.com/downloads/en/devicedoc/doc2467.pdf (дата обращения: 26.01.2025).

69. Godbolt, M. / M. Godbolt. – Текст : электронный // Compiler Explorer : [сайт]. – URL: https://godbolt.org/ (дата обращения: 26.01.2025).

70. Ranea, A. CASCADA / A. Ranea., V. Rijmen. – Текст : электронный // Github : [сайт]. – URL: https://github.com/ranea/CASCADA (дата обращения: 26.01.2025).

71. Zheng, X. Do NOT Misuse the Markov Cipher Assumption Automatic Search for Differential and Impossible Differential Characteristics in ARX Ciphers / X. Zheng, L. Yongqiang, J. Lin [и др.]. – Текст : непосредственный // Cryptology ePrint Archive. – 2022.

ПРИЛОЖЕНИЕ А. АКТ О ВНЕДРЕНИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ (НАУЧНОЕ НАПРАВЛЕНИЕ КАФЕДРЫ ИБТКС)

УТВЕРЖДАЮ
Директор Института
компьютерных технологий
и информационной безопасности
_____ Г. Е. Веселов
« 16 » января 2024 г.

Акт о внедрении результатов диссертационной работы
Прудникова Вадима Александровича
кафедры информационной безопасности телекоммуникационных систем

Диссертационные исследования аспиранта Прудникова Вадима Александровича связаны с инициативной научно-исследовательской работой «Метод синтеза псевдо-динамических операций подстановки с предельными криптографическими характеристиками для семейства перспективных легковесных псевдослучайных функций pCollapser», выполняемой в рамках научной тематики кафедры информационной безопасности телекоммуникационных систем (ИБТКС).

Результаты диссертационных исследований посвящены разработке структуры псевдо-динамической операции подстановки на основе ARX-функций, удовлетворяющей широкому спектру противоречивых требований, а также разработке программной реализации псевдо-случайной pCollapserARX на основе псевдо-динамических операций подстановки, использующих ARX-функции.

В отчётной документации и в выполнении плановых показателей по научной работе кафедры информационной безопасности телекоммуникационных систем института компьютерных технологий и информационной безопасности отражены публикации и участие в научных конференциях аспиранта Прудникова Вадима Александровича, где нашли отражение его наиболее существенные научные результаты:

– структура псевдо-динамической операции подстановки на основе ARX-функций, удовлетворяющая широкому спектру противоречивых требований;

– семейство псевдослучайных функций pCollapserARX, обладающих экстремальной параллелизацией обработки данных и предназначенных для применения в качестве высокопроизводительной PRF в режимах AEAD, CTR, Sponge-конструкций.

По результатам исследований опубликовано 9 научных работ. Из них в перечне рецензируемых научных изданиях, рекомендованных ВАК Минобрнауки России для публикации материалов диссертаций на соискание ученых степеней кандидата и доктора технических наук, опубликовано 3 статьи:

– Поликарпов С.В., Прудников В.А., Кожевников А.А., Румянцев К.Е. Псевдослучайная функция PCOLLAPSER, обеспечивающая экстремальный

параллелизм обработки информации. Известия ЮФУ. Технические науки. Издательство Южного федерального университета, 2019 №5 (207).

– Поликарпов С.В., Прудников В.А., Румянцев К.Е. Вычислительно эффективный метод определения усреднённых линейных свойств псевдо-динамических подстановок. Известия ЮФУ. Технические науки. 2020. №5 (215).

– Поликарпов С.В., Румянцев К.Е., Прудников В.А. Исследование свойств миниверсии псевдо-случайной функции pCollapser. Известия ЮФУ. Технические науки. Издательство Южного федерального университета, №6 (230). 2022 г., г. Ростов-на-Дону, 2022 г.

Публикации в реферируемых изданиях, учитываемых в РИНЦ:

– Прудников В.А. Исследование нелинейных свойств псевдодинамической подстановки PD-SBOX 6x4x4. Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности, Сборник статей V Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов. Ростов-на-Дону: Издательство ЮФУ, 2019.

– Поликарпов С.В., Прудников В.А., Румянцев К.Е. Программные инструменты анализа нелинейных свойств криптографических подстановок. I Всероссийская научно-практическая конференция «Digital Era» 26.03.2021, Издательство Чеченского государственного университета, г. Грозный, 2021 г.

– Прудников В.А. Программный инструмент анализа нелинейных характеристик криптографических подстановок, использующий многопоточные вычисления. Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности, Сборник статей VII Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов. Таганрог: Издательство ЮФУ, 2021.

– Прудников В.А. Анализ линейных свойств псевдо-динамической подстановки на базе ARX-конструкций. Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности, Сборник статей VIII Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов. Таганрог: Издательство ЮФУ, 2022.

– Прудников В.А. Анализ количества активных ARX-функций для миниверсии PRF pCollapser-ARX. Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности, Сборник статей IX Всероссийской научно-технической конференции молодых ученых, аспирантов, магистрантов и студентов. Таганрог: Издательство ЮФУ, 2023.

Результаты диссертационных исследований апробированы на международных и всероссийских научно-технических конференциях:

– V, VII–IX Всероссийские научно-технические конференции молодых ученых, аспирантов, магистрантов и студентов «Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности», 1–7.04.2019, 5–11.04.2021, 4–9.04.2022, 10–15.04.2023. Таганрог.

– I Всероссийская Научно-Практическая Конференция «Digital Era», 26 марта 2021, г. Грозный;

– XII симпозиум «Современные тенденции в криптографии» (CTCrypt 2023) 6-9 июня 2023, г. Волгоград).

Практическая ценность работы заключается в том, что использование псевдо-динамических подстановок на основе подобранных ARX-функций в структуре псевдослучайной функции pCollapser позволяет обеспечить значения весов дифференциальных характеристик, превосходящие аналоги в два раза, при тех же затратах ресурсов при программной реализации.

Зам. зав. кафедрой ИБТКС, к.т.н. С. Л. Балабаев
« 10 » января 2024 г.

Доцент кафедры ИБТКС, к.т.н. С. В. Поликарпов
« 10 » января 2024 г.

Доцент кафедры ИБТКС, к.т.н., доцент А. П. Плёнкин
« 10 » января 2024 г.

ПРИЛОЖЕНИЕ В. АКТ О ВНЕДРЕНИИ ПРОГРАММЫ ЭВМ

УТВЕРЖДАЮ
Директор Института
компьютерных технологий
и информационной безопасности
_____ Г. Е. Веселов
« 4 » апреля 2024 г.

Акт о внедрении программы ЭВМ
Прудникова Вадима Александровича в учебный процесс
кафедры информационной безопасности телекоммуникационных систем

Аспирант кафедры ИБТКС Прудников Вадим Александрович совместно с доцентом кафедры ИБТКС Поликарповым Сергеем Витальевичем под руководством заведующего кафедрой информационной безопасности телекоммуникационных систем Румянцева Константина Евгеньевича разработали программное обеспечение «Программа анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций».

Программа предназначена для анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций – дифференциальных свойств, линейных свойств, алгебраического иммунитета, количества степеней свободы.

Программа может быть использована организациями, обладающими необходимостью анализа криптографических свойств (дифференциальные свойства, линейные свойства, алгебраический иммунитет, количество степеней свободы) псевдо-динамических операций подстановки на основе ARX-конструкций. Результаты анализа могут быть использованы для разработки или модификации существующих криптоалгоритмов, а также псевдослучайных функций, с целью повышения криптографической стойкости и эффективности их работы (потребление ресурсов, производительность).

Программа ЭВМ предназначена для студентов специальности 10.05.02 «Информационная безопасность телекоммуникационных систем» и используется в дисциплине «Разработка программно-аппаратных средств телекоммуникационных систем», которая читается в 8-м семестре. Программа ЭВМ может быть полезна для подготовки дипломированных специалистов по специальностям и направлению укрупнённой группы 10.00.00 «Информационная безопасность».

Заведующему кафедрой ИБТКС Румянцеву Константину Евгеньевичу принадлежит общая постановка задачи по разработке программного обеспечения для анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций.

Аспиранту Прудникову Вадиму Александровичу принадлежит разработка программной реализации алгоритма анализа криптографических свойств псевдо-

динамических операций подстановки на основе ARX-конструкций, интерфейса программы, а также тестирование и отладка программной реализации.

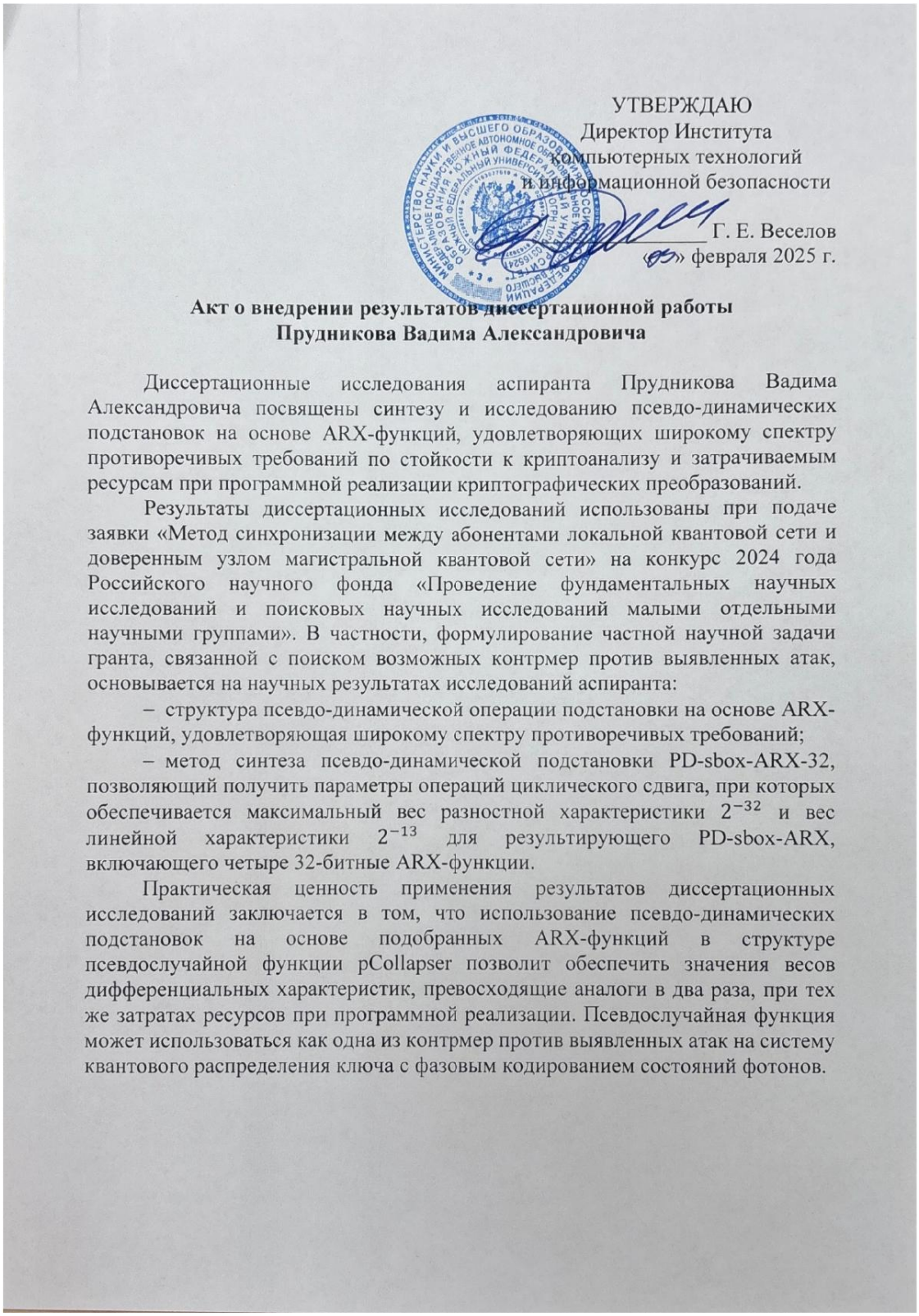
Доценту кафедры ИБТКС Поликарпову Сергею Витальевичу принадлежит разработка принципов анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций, а также разработка алгоритма анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций.

Зам. зав. кафедрой ИБТКС, к.т.н. С. Л. Балабаев
« 3 » апреля 2024 г.

Доцент кафедры ИБТКС, к.т.н., доцент А. В. Горбунов
« 3 » апреля 2024 г.

Доцент кафедры ИБТКС, к.т.н., доцент А. В. Помазанов
« 3 » апреля 2024 г.

ПРИЛОЖЕНИЕ С. АКТ О ВНЕДРЕНИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ



УТВЕРЖДАЮ
Директор Института
компьютерных технологий
и информационной безопасности
_____ Г. Е. Веселов
«05» февраля 2025 г.

Акт о внедрении результатов диссертационной работы
Прудникова Вадима Александровича

Диссертационные исследования аспиранта Прудникова Вадима Александровича посвящены синтезу и исследованию псевдо-динамических подстановок на основе ARX-функций, удовлетворяющих широкому спектру противоречивых требований по стойкости к криптоанализу и затрачиваемым ресурсам при программной реализации криптографических преобразований.

Результаты диссертационных исследований использованы при подаче заявки «Метод синхронизации между абонентами локальной квантовой сети и доверенным узлом магистральной квантовой сети» на конкурс 2024 года Российского научного фонда «Проведение фундаментальных научных исследований и поисковых научных исследований малыми отдельными научными группами». В частности, формулирование частной научной задачи гранта, связанной с поиском возможных контрмер против выявленных атак, основывается на научных результатах исследований аспиранта:

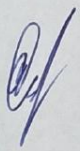
– структура псевдо-динамической операции подстановки на основе ARX-функций, удовлетворяющая широкому спектру противоречивых требований;

– метод синтеза псевдо-динамической подстановки PD-sbox-ARX-32, позволяющий получить параметры операций циклического сдвига, при которых обеспечивается максимальный вес разностной характеристики 2^{-32} и вес линейной характеристики 2^{-13} для результирующего PD-sbox-ARX, включающего четыре 32-битные ARX-функции.

Практическая ценность применения результатов диссертационных исследований заключается в том, что использование псевдо-динамических подстановок на основе подобранных ARX-функций в структуре псевдослучайной функции pCollapser позволит обеспечить значения весов дифференциальных характеристик, превосходящие аналоги в два раза, при тех же затратах ресурсов при программной реализации. Псевдослучайная функция может использоваться как одна из контрмер против выявленных атак на систему квантового распределения ключа с фазовым кодированием состояний фотонов.

Заявка «Метод синхронизации между абонентами локальной квантовой сети и доверенным узлом магистральной квантовой сети» вошла в список победителей на конкурсе 2024 года Российского научного фонда «Проведение фундаментальных научных исследований и поисковых научных исследований малыми отдельными научными группами».

Руководитель гранта РНФ
доцент кафедры ИБТКС,
к.т.н., доцент



А. П. Плёнкин
«03» февраля 2025 г.

ПРИЛОЖЕНИЕ D. СВИДЕТЕЛЬСТВО О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ ПРОГРАММЫ ДЛЯ ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024610931

Программа анализа криптографических свойств псевдо-динамических операций подстановки на основе ARX-конструкций

Правообладатель: ***федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет» (RU)***

Авторы: ***Поликарпов Сергей Витальевич (RU), Прудников Вадим Александрович (RU), Румянцев Константин Евгеньевич (RU)***

Заявка № **2023688912**

Дата поступления **21 декабря 2023 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***16 января 2024 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429b6a0fe3853164baf96f83b73b4aa7
Владелец Зубов Юрий Сергеевич
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов