УДК 004.056 **DOI** 10.15622/ia.24.2.6

Д.В. НАГИБИН, А.С. ПЕТРЕНКО, В.С. ДАВЫДЕНКО, И.В. КОТЕНКО, Е.В. ФЕДОРЧЕНКО

ИНВАЗИВНЫЙ ПОДХОД К ВЕРИФИКАЦИИ ФУНКЦИОНАЛЬНО-СТРУКТУРНЫХ СПЕЦИФИКАЦИЙ, РЕАЛИЗОВАННЫХ В ЗАКАЗНЫХ ИНТЕГРАЛЬНЫХ СХЕМАХ

Нагибин Д.В., Петренко А.С., Давыденко В.С., Котенко И.В., Федорченко Е.В. **Инвазивный подход к верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах.**

Аннотация. Представлен подход к верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах, основанный на инвазивных методах исследования. Актуальность проведённого исследования обусловлена необходимостью проведения верификации функционально-структурных спецификаций, поставляемых сторонними исполнителями аппаратных реализаций алгоритмов обеспечения информационной безопасности, сложностью выявления на аппаратном уровне модификаций этих алгоритмов и внедрённых в них недокументированных возможностей и отсутствием единых универсальных или стандартизированных методов решения этой задачи. Сформулирована математическая постановка задачи исследования, суть которой состоит в проверке равенства значений параметров заявленной спецификации с их значениями, восстановленными методом обратного проектирования. Представлены результаты применения предложенного подхода к верификации функционально-структурных спецификаций на примерах аппаратно-реализованных алгоритмов шифрования DES и AES. Восстановленные функционально-структурные блоки алгоритмов (в частности – блок подстановок) были успешно верифицированы.

Ключевые слова: заказная интегральная схема, идентификация, верификация, функционально-структурная спецификация, алгоритм шифрования.

1. Введение. Стойкость криптографических алгоритмов определяется используемым ключом шифрования, который для нарушителя изначально неизвестен. Сами алгоритмы шифрования и их параметры открыты для ознакомления, изучения, проверок и поисков уязвимостей. При этом внесение любого изменения в криптографические алгоритмы может привести к появлению уязвимостей и, как следствие, к снижению стойкости алгоритма. Поэтому проведение проверок в поставляемом оборудовании или программном обеспечении на предмет соответствия реализованных в них криптографических алгоритмов заявленным спецификациям является актуальной задачей для информационной безопасности [1, 2].

Особенно остро эта задача стоит при верификации алгоритмов, реализованных аппаратно, так как, по сравнению с программными реализациями, аппаратные реализации предоставляют гораздо больше вариантов для скрытного внесения модификаций и недокументированных возможностей в технические устройства.

Указанная задача декомпозируется на 1) идентификацию алгоритмов в исследуемом устройстве и 2) их верификацию.

Решение первой из указанных задач было предложено в статье [1]. В данной работе рассматривается верификация алгоритмов на примерах исследования аппаратно-реализованных алгоритмов шифрования DES (Data Encryption Standard) [3 – 5] и AES (Advanced Encryption Standard) [6 – 8], который де-факто является международным стандартом шифрования.

Хотя попытки автоматизировать решение задачи верификации программных и/или аппаратных систем в мировом сообществе ведутся давно, на сегодняшний день отсутствуют единые универсальные или стандартизированные методы решения этой задачи. Связано это с разнородностью реализаций, осуществляемых различными производителями. Кроме того, трудность этой задачи обусловлена и архитектурой построения современных технических устройств, целевая функция которых распределяется между программной и аппаратной реализациями. Само это распределение не стандартизировано. Таким образом, для решения задачи верификации технических устройств или систем необходимо проводить исследования как программных, так и аппаратных частей их реализаций.

Настоящая работа направлена на решение задачи верификации функционально-структурных спецификаций аппаратных реализаций технических устройств. Данная работа отличается применением инвазивных методов исследования и нацеленностью на анализ аппаратных реализаций алгоритмов шифрования, что определило некоторые её особенности. Исследование вносит вклад в практику решения проблемы верификации функционально-структурных спецификаций заказных интегральных схем (ЗИС) недоверенных производителей, например, иностранного производства.

При этом термин «спецификация» в разных отраслях определён по-разному. В рамках настоящего исследования будем руководствоваться следующей формулировкой: спецификация – это документ, содержащий подробное перечисление узлов и деталей какого-либо изделия, конструкции, установки, и т.п., входящих в состав сборочного или монтажного чертежа, а также документ с перечислением условий, которым должен удовлетворять производственный заказ.

Работа организована следующим образом. В разделе 2 дается общая характеристика релевантных исследований. В разделе 3 формулируется задача исследования. В разделе 4 предлагается подход

к верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах. В разделе 5 описывается применение данного подхода на примере алгоритма шифрования DES. В разделе 6 рассматривается применение подхода к верификации на примере алгоритма шифрования AES. В разделе 7 оценивается результативность применения этого подхода. В разделе 8 подводятся итоги проделанной работы и определяются направления дальнейших исследований.

Алгоритмы DES и AES выбраны, потому что мировым сообществом они признаны в качестве стандартов шифрования. В дальнейшем планируется применение предложенных решений к другим аппаратно реализованным алгоритмам.

2. Релевантные исследования. Можно выделить следующие исследования, посвященные автоматизации решения задачи верификации программных и/или аппаратных систем.

В [9 – 11] представлены общие принципы проектирования защищенных систем, включая рассмотрение вопросов, связанных с верификацией различных классов систем: систем на основе микроконтроллеров [9], защищенных встроенных устройств на примере системы охраны периметра [10] и безопасных встроенных систем [11].

Ряд работ направлен на статическую верификацию исполняемых программ и исходного кода программ. В.В. Ковалев и др. [12] предложили подход к статической верификации исполняемых программ, основанный на сопоставлении семантических аспектов вычислений. В [13] рассматривается статический анализ исходного кода программ на основе абстрактной интерпретации. В [14] авторы разработали подход к верификации программ с «нулевыми» знаниями, т.е. без раскрытия исходного кода.

В [15] авторы рассмотрели подход к верификации протоколов безопасности на основе комбинированного использования существующих методов и средств. Tran D.D. и Ogata K. разработали инвариантный генератор оценки доказательства IPSG [16] на основе интерпретатора CafeOBJ [17, 18]. В [19] описано его применение для верификации протокола TLS. Saeed и др. предложили подход к формальной верификации криптографических протоколов с использованием методов проверки на модели и частичного порядка [20]. В [21] авторы разработали инструмент CryptoFormalEval автоматического выявления уязвимостей в криптографических протоколах на основе больших языковых моделей и формальной верификации.

В [22] авторы предложили метод верификации функционально-структурных спецификаций технических устройств на основе интерпретатора CafeOBJ с поддержкой тестовых оценок и проверкой теорем, а также генератором тестовых сценариев на основе тестовых оценок с возможностью исправления ошибок.

В [23] авторы разработали метод установления эквивалентности скомпилированных двоичных файлов для конкретного программного обеспечения на основе проверки того, являются ли исходный и двоичный графы потока управления изоморфными после преобразований графа, сохраняющих семантику.

В [24] авторы представили метод адаптивного извлечения структурных признаков аппаратных реализаций троянских программ на уровне шлюза GateDet, который включает в себя усовершенствованный метод моделирования графов схем и разработанный метод обнаружения, основанный на двунаправленных графовых нейронных сетях.

В [25] авторы предложили метод идентификации аппаратно-реализованных троянов на основе анализа данных из побочных каналов с применением сетевых архитектур ResNeXt и ARA, позволяющий эффективно различать типы аппаратных троянских программ, в том числе и в условиях отсутствия эталонного образца.

В [26] авторы разработали модель процесса обратного проектирования сложных производственных систем, которая представляет собой сложную систему динамических, итеративных, параллельных, рекурсивных и зависящих от времени процессов.

В [27] авторы разработали многопараметрическую систему обнаружения троянских программ на основе машинного обучения с поддержкой анализа логики передачи регистров.

В [28] авторы предложили неинвазивный метод обнаружения аппаратно-реализованных троянских программ, основанный на использовании аппаратных средств с нулевой задержкой (с использованием пространственных корреляций для подавления методов скрытия троянских программ).

В [29] авторы рассмотрели методику преобразования программного обеспечения, направленную на внедрение обфускации при выполнении программ. Целью исследования являлась защита микропроцессорных систем от недокументированных возможностей аппаратных реализаций на основе создания модифицированной версии ассемблерного кода.

В [30] авторы описали методику идентификации аппаратно-реализованных троянов в конвейерных микропроцессорах на основе использования семейства классификаторов.

В [31] авторы представили методику идентификации аппаратно-реализуемых троянов, внедряемых в аппаратное обеспечение средствами систем автоматизированного проектирования, на основе методов машинного обучения.

Недостатком проанализированных исследований является применение неинвазивных методов, которые нацелены, в основном, именно на поиск недокументированных возможностей. К таким методам относится анализ побочных каналов, пространственная корреляция, машинное обучение, анализ управляющих программ и др. Использование только процедуры поиска не позволяет говорить однозначно о гарантированном отсутствии недокументированных возможностей.

Настоящая работа применяет инвазивные (разрушающие) методы исследования и имеет более общую цель – верификацию функционально-структурных спецификаций технических устройств, которая включает также выявление недокументированных возможностей. Специфической особенностью предлагаемого подхода является нацеленность на анализ аппаратных реализаций алгоритмов шифрования.

3. Постановка задачи исследования. Нелинейность криптографическим преобразованиям в блочных шифрах придают блоки подстановок (так называемые S-блоки, от английского «substitution»), что определило высокий интерес специалистов в области информационной безопасности к анализу этих блоков.

Определение 1. S-блок (S-box, блок замены, блок подстановки) – представляет собой отображение из множества двоичных векторов длины n в множество двоичных векторов длины m [32].

При этом числа n и m относительно малы, например 4, 6, 8, 16, 32. Часто рассматриваются S-блоки, являющиеся взаимно однозначными преобразованиями, в этом случае $n = m$.

Обычно они хранятся в виде таблиц как массивы данных. На рисунке 1 представлен фрагмент схемы криптографического преобразования информации с помощью S-блоков в алгоритме DES [33], который можно считать типовым для подобных преобразований. Структура криптографического преобразования с помощью S-блоков алгоритма DES насчитывает 8 таких блоков. Кроме того, на ней наглядно проиллюстрирована разноразрядность

входного и выходного блока. Однако эти параметры в других алгоритмах могут отличаться.

Обычно входные данные для S-блоков рассматриваются как адрес ячейки S-блока, значение которой считывается и подаётся в качестве выходной информации из S-блока.

Рис. 1. Схема криптографического преобразования информации с помощью S-блоков в алгоритме DES

Математически S-блок является векторной булевой функцией.

Определение 2. Векторная булева функция – функция вида

$$F : \mathbb{Z}_2^n \to \mathbb{Z}_2^m .$$

Другое название векторной булевой функции: (n,m)-функция [32].

На рисунке 2 представлены шестнадцатеричные значения блока подстановок алгоритма AES [6].

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

Рис. 2. Значения блока подстановок алгоритма AES

В рассматриваемом примере разрядность входных данных равна 8 битам (4 бита определяют адрес строки, 4 бита определяют адрес столбца в таблице) и размерность выходных данных равна 8 битам (определяется количеством бит, необходимых для представления шестнадцатеричных чисел от 0x00 до 0xFF).

Самый распространённый вариант хранения этой таблицы – хранение в памяти. Однако блоки памяти можно считывать и перезаписывать. В этом случае довольно проблематично скрытно внести в них модификации. То есть, такой вариант хранения значений блоков подстановок прост для реализации, но и прост для верификации этих значений.

Второй вариант представления подобных таблиц подстановок – в виде булевых функций каждого выходного бита от входных битов.

Определение 3. Произвольная функция f из множества $\mathbb{Z}_2^n$ в множество $\mathbb{Z}_2$ называется булевой функцией от n переменных [33].

Приведём пример булевой функции от шести переменных: $f\left(x_5, x_4, x_3, x_2, x_1, x_0\right) = \left(x_5 \cdot x_3 \cdot x_1\right) \oplus \left(x_4 + x_2 + x_0\right)$, где $x_0, \ldots, x_5 \in \mathbb{Z}_2$.

Если этот вариант реализован аппаратно, то для его проверки (верификации) необходимо использовать средства аппаратного обратного проектирования. Внедрённые модификации на аппаратном уровне могут долгое время оставаться не выявленными для конечных пользователей технических устройств, что (как ранее было сказано во введении) является уязвимостью, через которую могут быть реализованы различные угрозы информационной безопасности (получение доступа к закрытой переписке, искажение информации и прочее) [34, 35].

Следует отметить, что таблицы подстановок для каждого алгоритма шифрования подвергаются тщательному изучению и проверке на уязвимости мировым криптографическим сообществом [36]. Потому в основных криптографических алгоритмах и стандартах значения таблиц S-блоков не рекомендуется изменять, так как их исходные значения уже прошли указанные проверки и общепризнаны стойкими к различным методам криптоанализа. Считается, что любое изменение значений таблиц подстановок (как и перестановок) может существенно снизить стойкость алгоритма шифрования к атакам на него [37].

Таким образом, задачу верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах, можно сформулировать следующим образом.

Дано:

1. X_{Pr} – исходный файл с описанием элементной базы исследуемой ЗИС, представленной в виде помеченного связного графа [1];

2. Z_M – база знаний интеллектуальной системы поддержки принятия решений (ИСППР), содержащая знания о спецификациях известных и восстановленных ЗИС [1];

3. $U^{SpO} = \left\{u_1^{SpO}, u_2^{SpO}, \ldots, u_e^{SpO}\right\}$ – множество спецификаций, потенциально реализуемых на знаниях Z_M [1];

4. $U^{SpC} = \left\{u_1^{SpC}, u_2^{SpC}, \ldots, u_c^{SpC}\right\}$ – множество спецификаций, потенциально реализуемых на основе восстановленной структуры ЗИС [1].

На основе анализа исходного файла X_{Pr} и ИСППР Z_M, содержащей знания о спецификациях известных и восстановленных ЗИС, сформируем универсальный для каждого криптографического алгоритма, содержащего блоки подстановок, набор параметров:

$$P = \left\{n, r, m; c, d\right\}, \tag{1}$$

где:

n – разрядность входного блока;

r – количество S-блоков;

m – разрядность выходного блока;

c – разрядность входных данных на каждом S-блоке;

d – разрядность выходных данных на каждом S-блоке.

Найти (проверить): рассматриваемая задача верификации сводится к проверке равенства:

$$u_i^{SpO} = u_j^{SpC}, \tag{2}$$

где:

u_i^{SpO} – спецификация исследуемого устройства согласно поставляемой производителем технической документации;

u_j^{SpC} – спецификация исследуемого устройства, которая формируется в ходе проведения исследований на основе значений выражения (1) и значений, полученных при выявлении полного

множества значений исследуемых блоков подстановок методом полного перебора всего диапазона их адресов.

Выражение (2) будет выполняться при выполнении следующего равенства:

$$P_{Спец} = P_{Иссл}, \tag{3}$$

где:

$P_{Спец}$ – значения параметров блоков подстановок согласно заявленной спецификации исследуемого устройства;

$P_{Иссл}$ – выявленные значения параметров блоков подстановок исследуемого устройства.

4. Подход к верификации. Структурная схема алгоритма верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах, представлена на рисунке 3.

Блокам 2 и 3 рисунка 3 соответствуют пункты 1 – 4 и формирование выражения (1) математической постановки задачи.

Так как входные данные рассматриваются как адреса ячеек в S-блоках (значения которых необходимо подать в качестве выходных данных), то для идентификации каждого S-блока предлагается анализ считываемых значений этих блоков по следующим адресам:

- адрес нулевой ячейки (на вход каждого блока подстановок подаются все нули);
- старший адрес в таблице (на вход каждого блока подстановок подаются все единицы);
- так называемое «протягивание» в адресе считываемой ячейки единицы (количество считанных таким образом ячеек равно размерности входных данных блока подстановок c);
- так называемое «протягивание» в адресе считываемой ячейки нуля (количество считанных таким образом ячеек равно размерности входных данных блока подстановок c).

Количество a считанных для анализа ячеек в каждом блоке подстановок равно

$$a(S_i) = 2 + 2 \cdot c, \tag{4}$$

где:

S_i – тестируемый блок подстановок и $i \in [1; r]$;

r – количество S-блоков;

c – разрядность входных данных на каждом S-блоке.

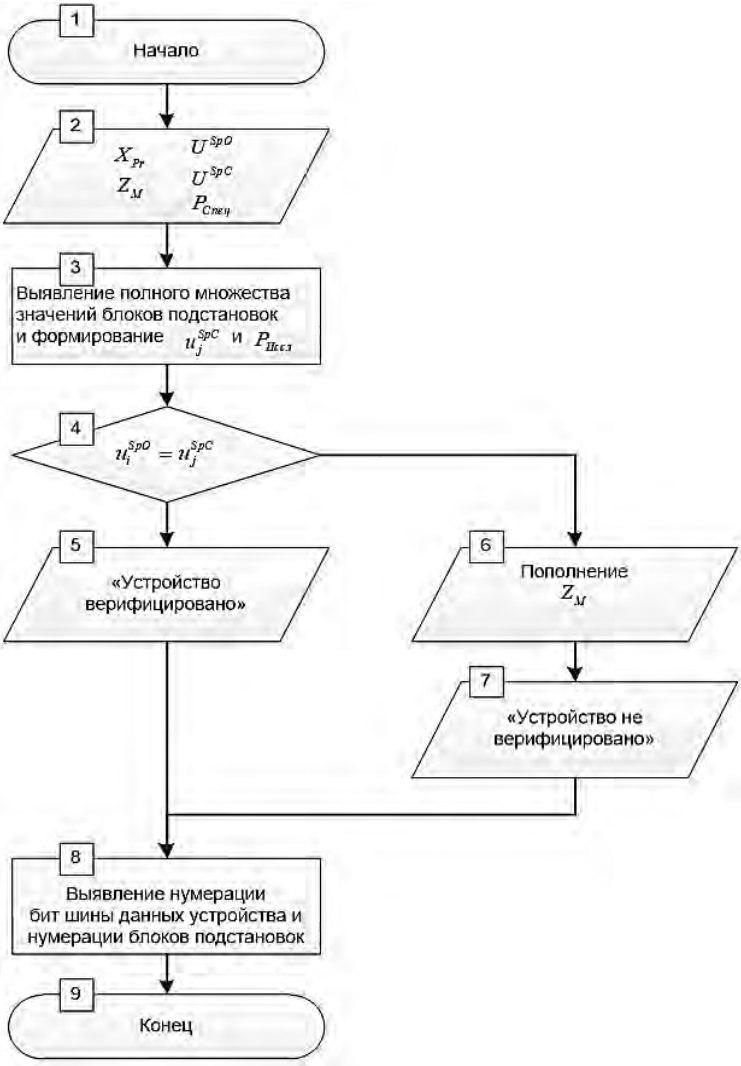

Рис. 3. Структурная схема алгоритма верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах

Общее количество считанных для анализа ячеек из всех блоков подстановок представлено выражением:

$$\sum_{i=1}^{r} a\left(S_i\right) = 2 \cdot r \cdot \left(1 + c\right). \quad (5)$$

В общем случае количество блоков подстановок r в алгоритмах шифрования больше единицы. Поэтому сначала проводится «первичная» идентификация исследуемого блока – это осуществляется указанным «протягиванием» единицы и нуля. Анализ исследованных алгоритмов показал, что во всех случаях применения такого подхода идентификация проходила успешно.

Такой результат был получен и при анализе указанного алгоритма DES (результаты представлены в разделе 5).

В случае алгоритма AES можно сразу переходить к процессу верификации, так как в AES имеется только один блок подстановок (раздел 6).

Тестируемый алгоритм может оказаться известным и содержаться в базе знаний ИСППР Z_M. В этом случае переходим к процессу верификации тестируемых блоков подстановок с их каноническими значениями из имеющейся базы знаний – проверке равенств (2) и (3).

В случае подтверждения равенства (2), делается вывод о *верификации* исследуемого устройства (или части устройства) его заявленным производителем спецификациям (блок 5 рисунка 3), а также вывод о доверии исследуемому устройству с точки зрения информационной безопасности.

В противном случае переходим к блоку 6 рисунка 3 и пополняем базу знаний ИСППР Z_M новым алгоритмом или выявленными модификациями уже известных алгоритмов. Далее делается вывод о *несоответствии* исследуемого устройства (или части устройства) его заявленным производителем спецификациям (блок 7 рисунка 3). Следовательно – исследуемое устройство не может быть признано доверенным для использования с точки зрения информационной безопасности.

В блоке 8 рисунка 3 происходит выявление нумерации S-блоков и выявление нумерации бит шины данных исследуемого алгоритма, что является важным при исследовании аппаратно реализованных алгоритмов шифрования. В начале исследования такая информация

отсутствует, как и значения параметров блоков подстановок, указанных в выражении (1).

На рисунке 4 представлена общая схема использования блоков подстановок в алгоритмах блочного шифрования, на которой проиллюстрировано отсутствие информации о параметрах и характеристиках этих алгоритмов.

Стоит заметить, что в алгоритме DES параметр r (количество S-блоков) имеет значение 8, а в алгоритме AES этот параметр равен 1 – для этого случая, конечно, выявление нумерации блоков подстановок не требуется.

Рис. 4. Общая схема использования блоков подстановок в блочных шифрах при отсутствии информации об их параметрах и характеристиках

Предполагается, что исследование проводится в условиях отсутствия информации о структуре анализируемых технических устройств.

5. Применение предложенного подхода к блокам подстановок алгоритма DES. Результат применения предложенного подхода представлен на примере блока подстановок S_2 алгоритма шифрования DES.

Согласно представленной в [1] методике идентификации реализованных в заказных интегральных схемах функционально-структурных спецификаций проводится обратное проектирование исследуемого функционально-структурного блока с укрупнениями (объединениями) элементов исследуемой интегральной схемы по предложенным уровням укрупнения при восстановлении функционально-

структурных спецификаций ЗИС. Указанная методика состоит из следующих шагов:

1. сбор и анализ первичной информации об исследуемой технике и ЗИС;
2. обратное проектирование заказной интегральной схемы;
3. выявление множества возможных режимов функционирования исследуемой заказной интегральной схемы.

Таким образом, настоящая работа является логическим продолжением работы [1] и заключается в переходе от процесса идентификации функционально-структурных блоков ЗИС к процессу их верификации.

На рисунке 5 схематично представлен пример восстановления одного логического блока. Из исходных данных в виде простейших логических элементов (блок 1 рисунка 5) происходит выделение задействованных элементов для создания нового более крупного элемента (блок 2 рисунка 5).

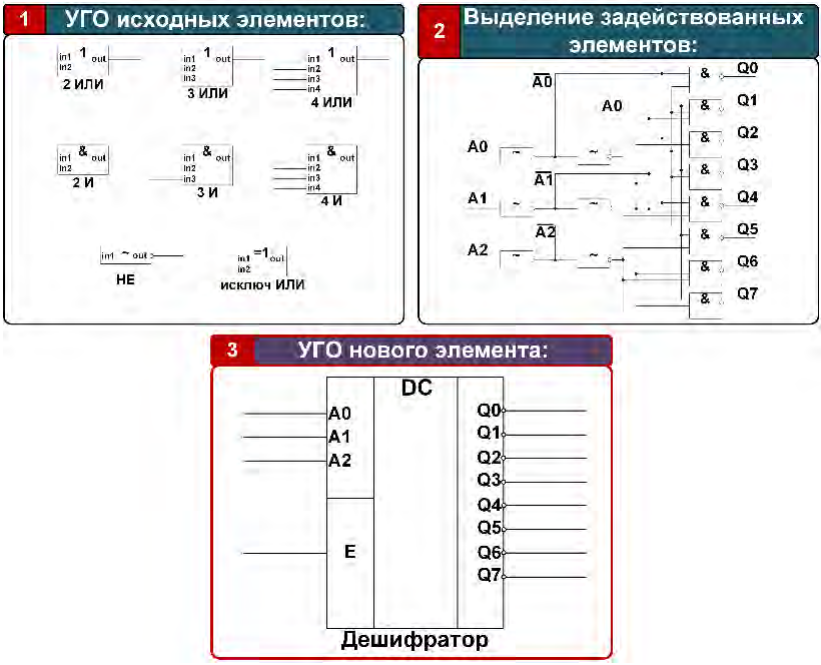

Рис. 5. Пример восстановления логического блока «дешифратор»

Далее проводится анализ логики функционирования выделенной группы элементов, на основе чего формируется условно-графическое обозначение (УГО) нового элемента – дешифратора трёх входных сигналов (блок 3 рисунка 5), который управляет обращением к одному из восьми блоков подстановок алгоритма DES.

Вновь образованный элемент (дешифратор) соответствует уровню 4 укрупнения функционально-структурных элементов заказных интегральных схем [1], что продемонстрировано на рисунке 6.

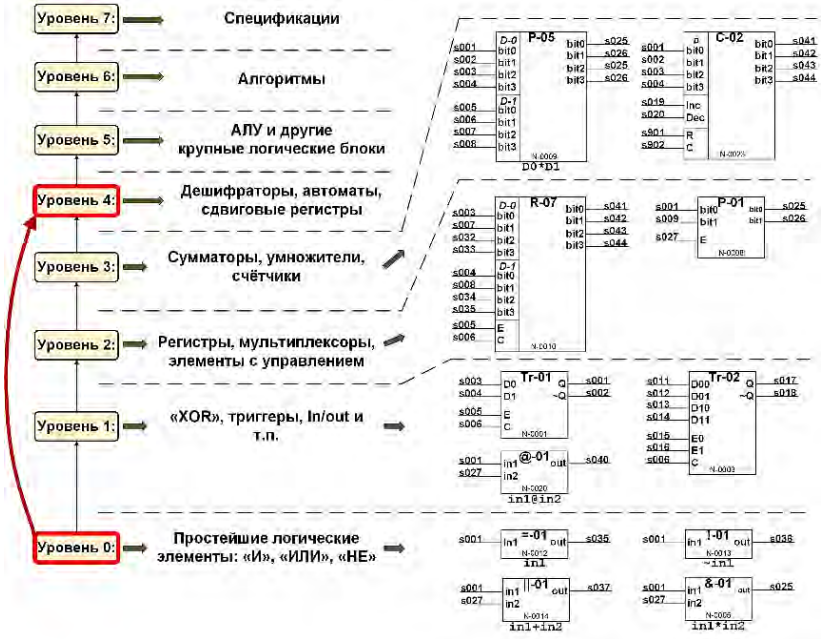

Рис. 6. Укрупнение логических элементов в функционально-структурные блоки

Пример применения предложенного подхода для блока подстановок S_2 алгоритма шифрования DES показан на рисунке 7.

Слева на рисунке 7 в таблице приведены значения тестируемого блока подстановок S_2 с представлением в двоичном виде адресов ячеек. Адреса строк в диапазоне [0;3], адреса столбцов в диапазоне [0;15].

Справа на рисунке 7 приведена таблица адресов считываемых ячеек блока подстановок S_2.

Результаты исследований на основе анализа блоков подстановок алгоритма DES показали, что считывание значений ячеек с указанных адресов позволяют однозначно идентифицировать тестируемые блоки подстановок с каноническими блоками подстановок алгоритма DES [3].

Предложенный подход к идентификации основан на анализе, согласно выражению (4), 14-ти из 64-х значений каждого блока подстановок алгоритма шифрования DES.

Как видно из рисунка 7, для анализа считываются по 5 значений ячеек из старшей и младшей строки, и по 2 значения ячеек (младший и старший столбец) из строки с номером $1_{10}=01_2$ и предпоследней строки (в рассмотренном примере – строка с номером $2_{10}=10_2$).

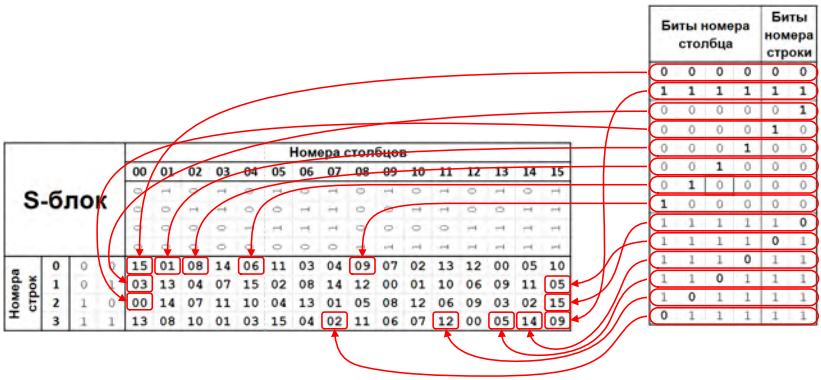

Рис. 7. Пример применения предлагаемого подхода к блоку подстановок S_2 алгоритма шифрования DES

Предложенный авторами подход не зависит от наличия у исследователя информации о нумерации бит на входе в блоки подстановок, так как «протягивание» сначала единицы и потом нуля по входным данным происходит по всем их разрядам. Эта нумерация, как и нумерация битов выходных значений блоков подстановок, была восстановлена в ходе проведения исследований.

В ходе исследования были проанализированы (согласно выражению (5)) 112 значений блока подстановок алгоритма шифрования DES. Как видно из рисунка 7, для анализа считываются по 5 значений ячеек из старшей и младшей строки, а также по 2 значения ячеек из младшего и старшего столбцов.

Таким образом, для рассмотренного примера выражения (1) и (5) имеют вид:

$$P = (n, r, m; c, d) = (48, 8, 32; 6, 4), \qquad \sum_{i=1}^{r} a(S_i) = 2 \cdot r \cdot (1 + c) = 112 .$$

По считанным указанным способом значениям происходит первичная идентификация блока подстановок.

Далее происходит последовательное считывание каждой ячейки блоков подстановок с верификацией исследуемых блоков подстановок или с пополнением базы данных алгоритмов шифрования (в случае, когда исследуемый алгоритм не идентифицирован как известный).

6. Применение разработанного подхода к S-блоку алгоритма AES. Результат применения предложенного подхода для S-блока алгоритма шифрования AES показан на рисунках 8 и 9.

Согласно [1] проводится обратное проектирование исследуемого функционально-структурного блока с укрупнениями (объединениями) элементов исследуемой ЗИС при восстановлении функционально-структурных спецификаций ЗИС по уровням укрупнения, представленным на рисунке 6. От «Уровня 0» (который соответствует простейшим логическим элементам «И», «Или», «Не») до «Уровня 4» (дешифраторы, автоматы, сдвиговые регистры).

Результат работы отображен на рисунке 8.

В верхней части рисунка 8 представлена логически связанная группа простейших логических элементов исследуемой ЗИС. Согласно структуре уровней детализации элементов (рисунок 6), образуемых в процессе восстановления функционально-структурных спецификаций заказных интегральных схем, на рисунке 8 (часть 1) представлены элементы, которые соответствуют нулевому уровню этой структуры. Анализ взаимосвязей локализованной группы элементов позволил создать новый функционально-логический элемент более высокого уровня укрупнения (структурно и логически представлен внизу на рисунке 8). Вновь образованный элемент (S-блок алгоритма AES) соответствует уровню 4 детализации.

В свою очередь, восстановленный блок подстановок является структурным элементом функции SubBytes (пятый уровень на рисунке 6) алгоритма AES (шестой уровень на рисунке 6).

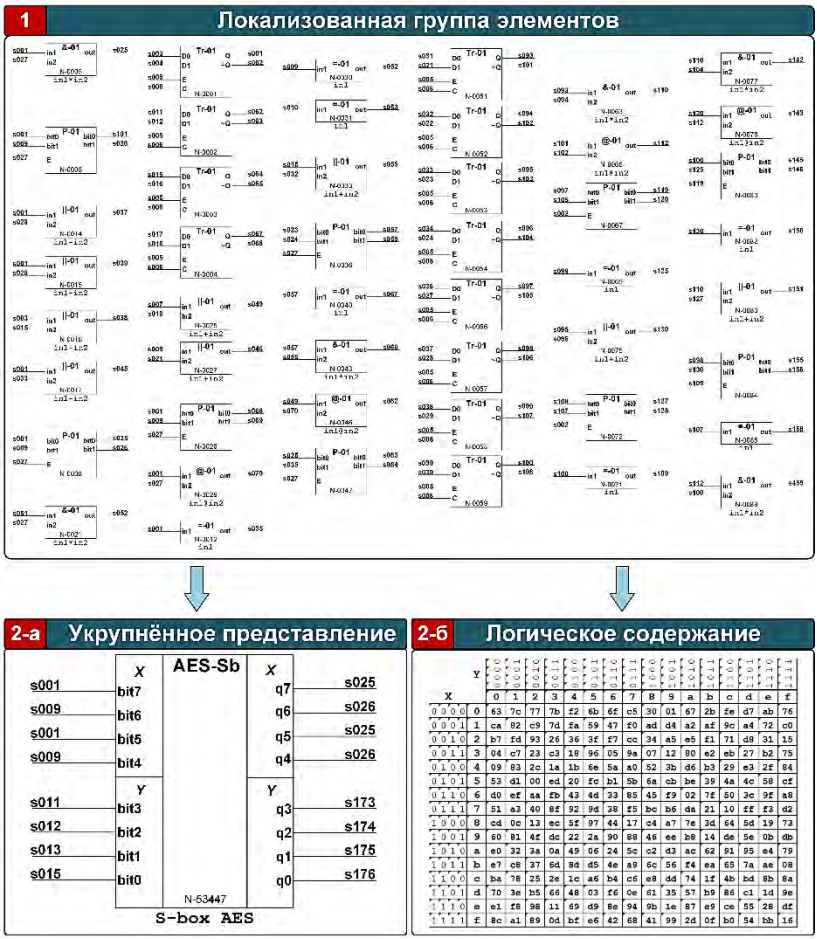

Рис. 8. Пример проведения укрупнения S-блока алгоритма шифрования AES

В верхней части рисунка 9 приведены значения проверяемого блока подстановок с представлением в двоичном виде адресов ячеек в таблице. Адреса строк в диапазоне [0;15], адреса столбцов в диапазоне [0;15].

В нижней части рисунка 9 приведена таблица с адресами и значениями считываемых ячеек исследуемого блока подстановок, которые также выделены в таблице верхней части рисунка.

X \ Y	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0000	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
0001	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
0010	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
0011	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
0100	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
0101	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
0110	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
0111	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
1000	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
1001	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
1010	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
1011	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
1100	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
1101	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
1110	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
1111	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

№ п.п.	Представление входных данных X	Y		Представление выходных данных X	Y
1	0 0 0 0	0 0 0 0	0 0	6 3	0 1 1 0 0 0 1 1
2	0 0 0 0	0 0 0 1	0 1	7 c	0 1 1 1 1 1 0 0
3	0 0 0 0	0 0 1 0	0 2	7 7	0 1 1 1 0 1 1 1
4	0 0 0 0	0 1 0 0	0 5	f 2	1 1 1 1 0 0 1 0
5	0 0 0 0	1 0 0 0	0 8	3 0	0 0 1 1 0 0 0 0
6	0 0 0 1	0 0 0 0	1 0	c a	1 1 0 0 1 0 1 0
7	0 0 1 0	0 0 0 0	2 0	b 7	1 0 1 1 0 1 1 1
8	0 1 0 0	0 0 0 0	4 0	0 9	0 0 0 0 1 0 0 1
9	1 0 0 0	0 0 0 0	8 0	c d	1 1 0 0 1 1 0 1
10	1 1 1 1	1 1 1 1	f f	1 6	0 0 0 1 0 1 1 0
11	1 1 1 1	1 1 1 0	f e	b b	1 0 1 1 1 0 1 1
12	1 1 1 1	1 1 0 1	f d	5 4	0 1 0 1 0 1 0 0
13	1 1 1 1	1 0 1 1	f b	0 f	0 0 0 0 1 1 1 1
14	1 1 1 1	0 1 1 1	f 7	6 8	0 1 1 0 1 0 0 0
15	1 1 1 0	1 1 1 1	e f	d f	1 1 0 1 1 1 1 1
16	1 1 0 1	1 1 1 1	d f	9 e	1 0 0 1 1 1 1 0
17	1 0 1 1	1 1 1 1	b f	0 8	0 0 0 0 1 0 0 0
18	0 1 1 1	1 1 1 1	7 f	d 2	1 1 0 1 0 0 1 0

Рис. 9. Пример применения предлагаемого подхода к S-блоку алгоритма шифрования AES

В ходе исследования были проанализированы (согласно выражению (5)) 18 значений блока подстановок алгоритма шифрования AES. Как видно из рисунка 9, для анализа считываются

по 5 значений ячеек из старшей и младшей строки, а также по 4 значения ячеек из младшего и старшего столбцов.

Таким образом, для рассмотренного примера выражения (1) и (5) имеют вид:

$$P = (n, r, m; c, d) = (8,1,8;8,8), \qquad \sum_{i=1}^{r} a\left(S_i\right) = 2 \cdot r \cdot \left(1 + c\right) = 18 .$$

По считанным указанным способом значениям происходит первичная идентификация блока подстановок.

Далее происходит последовательное считывание каждой ячейки блока подстановок с верификацией исследуемого блока подстановок или с пополнением базы данных алгоритмов шифрования (в случае, когда исследуемый алгоритм не идентифицирован как известный).

Проведённое авторами исследование не зависит от наличия информации о нумерации бит на входе в блок подстановок, так как «протягивание» сначала единицы и потом нуля по входным данным происходит по всем их разрядам. Эта нумерация, как и нумерация битов выходных значений блока подстановок, восстанавливается в ходе проведения исследований.

Результаты исследований на примере анализа блока подстановок алгоритма AES показали, что применение разработанного подхода позволяет однозначно верифицировать тестируемый блок подстановок с каноническим блоком подстановок алгоритма AES [6], что подтвердило работоспособность предложенных авторами решений.

7. Оценивание результативности применения подхода. Результаты применения предлагаемого подхода к исследованию аппаратных реализаций функционально-структурных элементов блочных алгоритмов шифрования для общего случая проиллюстрированы рисунком 10.

На рисунке 10 (в отличие от рисунка 4) показано, что в результате исследователю становятся известны параметры из выражения (1): размер входного блока *n*, количество S-блоков *r*, размер выходного блока *m*, размер входных данных на каждом S-блоке *c*, размер выходных данных на каждом S-блоке *d*. После чего проводится проверка равенства (4).

Кроме этого, в результате становится известна нумерация бит в шине данных, через которую происходит информационное взаимодействие других функционально-структурных элементов исследуемого алгоритма с блоками подстановок.

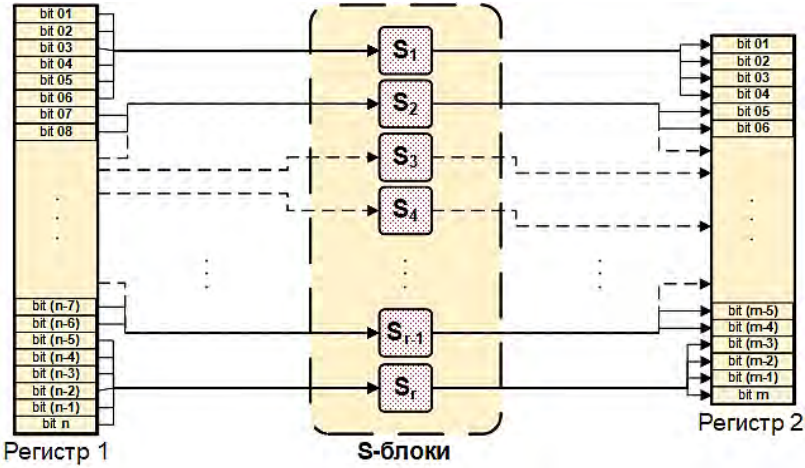

Рис. 10. Схема общего случая использования блоков перестановок в блочных шифрах с их выявленными параметрами и характеристиками

В таблице 1 представлены результаты сравнительного анализа предложенного авторами подхода и подходов, рассмотренных в разделе 2 настоящей статьи [9 – 15, 22 – 31].

Основной отличительной особенностью настоящего подхода является применение инвазивных методов исследования. Недостатком такого подхода является высокая стоимость, а достоинством – наличие полного доступа к анализу всех аппаратно-реализованных функционально-структурных блоков исследуемого устройства (в зависимости от назначения управляющей программы (так называемой «прошивки») – она может активировать, задействовать не полный перечень аппаратно-реализованного функционала устройства).

Строка 10 таблицы 1 показывает: разработанный авторами подход основан на применении методов онтологического представления знаний и разработки и применения правил для машины логического вывода, что позволяет выявлять блоки эквивалентных преобразований, конструировать новые правила для выявления подобных и других функционально-структурных блоков, использовать базу знаний исследованных ранее функционально-структурных спецификаций ЗИС. Проанализированные подходы методы онтологического представления знаний не применяют.

Таблица 1. Результаты сравнительного анализа разработанного подхода и релевантных работ

	Исследованные труды (нумерация согласно списку литературы) / **Применяемые методы**	[9]	[10]	[11]	[12]	[13]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	**Предложенный подход**
1.	Общие принципы проектирования защищённых систем	+	+	+													
2.	Методы машинного обучения									+		+			+	+	
3.	Методы тестовых оценок и генератора на их основе тестовых сценариев						+										
4.	Методы анализа обфусцированного программного кода													+			
5.	Методы сопоставления семантических аспектов вычислений				+												
6.	Методы анализа побочных каналов данных									+		+	+				
7.	Методы моделирования процесса обратного проектирования										+						
8.	Методы анализа САПР, которыми разрабатываются микросхемы											+				+	
9.	Методы моделирования и анализа графов исследуемых объектов				+			+	+								+
10.	Методы онтологического представления знаний и разработки и применения правил для машины логического вывода																+
11.	Учёт особенностей построения криптографических алгоритмов ЗИ											+					+
12.	Неинвазивные методы				+	+	+	+	+	+	+	+	+	+	+	+	
13.	Инвазивные методы																+

В статье [27] авторы рассмотрели в качестве примера применения своего подхода, основанного на методах анализа

побочных каналов данных (то есть – это тоже неинвазивные методы) и методах машинного обучения, один из режимов алгоритма AES. В отличие от такого подхода, настоящее исследование направлено на выявление особенностей аппаратных реализаций именно алгоритмов шифрования, иными словами – на создание модели функционально-структурных спецификаций аппаратных реализаций алгоритмов шифрования.

Проанализированные подходы представлены в таблице 1 в соответствии с применяемыми в них методами исследования. Стоит отметить, что в качестве примеров применения этих подходов были рассмотрены различные промышленные отрасли: авиастроение, автомобилестроение и другие; среди которых защиты информации криптографическими методами рассмотрено не было.

Интерес представляют исследования [27] и [31], которые рассматривают вредоносные аппаратные модификации, которые могут внедряться средствами систем автоматизированного проектирования (САПР) на этапе проектирования.

Таким образом, представленный подход может быть применён при проведении верификации функционально-структурных спецификаций или при проведении обратного проектирования интегральных микросхем. Особенностью настоящего подхода является его нацеленность на исследование аппаратных реализаций алгоритмов шифрования.

Применение данного подхода (как инвазивного метода исследования) в совокупности с неинвазивными методами даст полное представление о функционально-структурном наполнении исследуемого устройства, так как позволит выявить неиспользуемый управляющими программами аппаратно реализованный функционал и подтвердить или опровергнуть выдвинутые некоторыми статистическими неинвазивными методами гипотезы или выводы.

8. Заключение. Представлен подход к верификации функционально-структурных спецификаций, реализованных в заказных интегральных схемах.

Подтверждена работоспособность настоящего подхода на примерах исследования аппаратных реализаций алгоритмов шифрования DES и AES: начиная с процесса аппаратного обратного проектирования указанных алгоритмов (идентификации реализованных в заказных интегральных схемах функционально-структурных блоков) до верификации их функционально-логического наполнения.

Подтверждена работоспособность разработанного подхода для выявления нумерации исследуемых блоков подстановок и нумерации бит внутри каждого блока подстановки в отдельности.

Направлением будущей работы является проведение исследований по выявлению и локализации на стадии предварительных исследований функционально-структурных блоков, реализующих стандартные процессоры, восстановление и верификация которых может не представлять интереса.

Литература

1. Нагибин Д.В., Платонов А.А., Сабиров Т.Р. Методика идентификации реализованных в заказных интегральных схемах функционально-структурных спецификаций // Труды Военно-космической академии имени А.Ф. Можайского. 2024. № 690. С. 112–120.
2. Mustafa Dhiaa Al-Hassan, Qusay Zuhair Abdulla. Robust Password Encryption Technique with an Extra Security Layer // Iraqi Journal of Science. 2023. vol. 64. no. 3. pp. 1477–1486. DOI: 10.24996/ijs.2023.64.3.36.
3. Alsuwaiedi H.K.A., Rahma A.M.S. A new modified DES algorithm based on the development of binary encryption functions // Journal of King Saud University – Computer and Information Sciences. 2023. vol. 35(8). DOI: 10.1016/j.jksuci.2023.101716.
4. Agate V., Concone F., de Paola A., Ferraro P., Lo Re G., Morana M. Bayesian Modeling for Differential Cryptanalysis of Block Ciphers: A DES Instance // IEEE Access. 2023. vol. 11. pp. 4809–4820. DOI: 10.1109/ACCESS.2023.3236240.
5. Wu Y., Dai X. Encryption of accounting data using DES algorithm in Computing Environment // Journal of Intelligent & Fuzzy Systems. 2020. vol. 39. pp. 5085–5095. DOI: 10.3233/JIFS-179994.
6. Nitaj A., Susilo W., Tonien J. Enhanced S-boxes for the Advanced Encryption Standard with maximal periodicity and better avalanche property // Computer Standards & Interfaces. 2024. vol. 87. DOI: 10.1016/j.csi.2023.103769.
7. Zahid A.H., Arshad M.J. An innovative design of substitution-boxes using cubic polynomial mapping // Symmetry. 2019. vol. 11(3). DOI: 10.3390/sym11030437.
8. Zahid A.H., Arshad M.J., Ahmad M. Substitution-boxes using cubic fractional transformation // Entropy. 2019. vol. 21(3). DOI: 10.3390/e21030245.
9. Котенко И.В., Левшун Д.С., Чечулин А.А., Ушаков И.А., Красов А.В. Комплексный подход к обеспечению безопасности киберфизических систем на системе микроконтроллеров // Вопросы кибербезопасности. 2018. № 3(27). С. 29–38. DOI: 10.21681/2311-3456-2018-3-29-38.
10. Десницкий В.А., Чечулин А.А., Котенко И.В., Левшун Д.С., Коломеец М.В. Комбинированная методика проектирования защищенных встроенных устройств на примере системы охраны периметра // Труды СПИИРАН. 2016. № 5(48). С. 5–31. DOI: 10.15622/sp.48.1.
11. Desnitsky V., Levshun D., Chechulin A., Kotenko I. Design technique for secure embedded devices: application for creation of integrated cyber-physical security system // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). 2016. vol. 7. pp. 60–80.
12. Ковалев В.В., Компаниец Р.И., Новиков В.А. Верификация программ на основе соотношений подобия // Труды СПИИРАН. 2015. vol. 1(38). С. 233–245. DOI: 10.15622/sp.38.13.

13. Cousot P., Cousot R. A Galois connection calculus for abstract interpretation // Proceedings of the 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'14). ACM, 2014. pp. 3–4. DOI: 10.1145/2535838.2537850.
14. Fang Z., Darais D., Near J.P., Zhang Y. Zero Knowledge Static Program Analysis // Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS'21). New York, USA: Association for Computing Machinery, 2021. pp. 2951–2967. DOI: 10.1145/3460120.3484795.
15. Котенко И.В., Резник С.А., Шоров А.В. Верификация протоколов безопасности на основе комбинированного использования существующих методов и средств // Труды СПИИРАН. 2009. № 8. С. 292–310. DOI: 10.15622/sp.8.14.
16. Tran D.D., Ogata K. IPSG: Invariant Proof Score Generator // 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). 2022. pp. 1050–1055. DOI: 10.1109/COMPSAC54236.2022.00164.
17. Riesco A., Ogata K., Futatsugi K. A Maude environment for cafeOBJ // Formal Aspects of Computing. 2017. vol. 29. pp. 309–334. DOI: 10.1007/s00165-016-0398-7.
18. Clavel M., Durán F., Eker S., Lincoln P., Martí-Oliet N., Meseguer J., Talcott C. All about Maude – a High-Performance Logical Framework. How to Specify, Program and Verify Systems in Rewriting Logic // Lecture Notes in Computer Science. 2007. 802 p. DOI: 10.1007/978-3-540-71999-1.
19. Tran D.D., Ogata K. Formal verification of TLS 1.2 by automatically generating proof scores // Computers & Security. 2022. vol. 123. DOI: 10.1016/j.cose.2022.102909.
20. Saeed Mahlaqa, Ibrar Muhammad, Mahmood Dua, Delshadi Amirmohammad, Saleemi Aqdas. Enhancing Computer Security through Formal Verification of Cryptographic Protocols Using Model Checking and Partial Order Techniques // The Asian Bulletin of Big Data Management. 2024. vol. 4(2). pp. 225–238. DOI: 10.62019/abbdm.v4i02.176.
21. Curaba C., D'Ambrosi D., Minisini A., Antol'in N.P.-C. CryptoFormalEval: Integrating LLMs and Formal Verification for Automated Cryptographic Protocol Vulnerability Detection. 2024. arXiv preprint arXiv:2411.13627.
22. Riesco A., Ogata K. An integrated tool set for verifying CafeOBJ specifications // Journal of Systems and Software. 2022. vol. 189. DOI: 10.1016/j.jss.2022.111302.
23. Awadhutkar P., Tamrawi A., Goluch R., Kothari S. Control flow equivalence method for establishing sanctity of compiling // Computers & Security. 2022. vol. 115. DOI: 10.1016/j.cose.2022.102608.
24. Cheng D., Dong Ch., He W., Chen Zh., Liu X., Zhang H. A fine-grained detection method for gate-level hardware Trojan based on bidirectional Graph Neural Networks // Journal of King Saud University – Computer and Information Sciences. 2023. vol. 35(10). DOI: 10.1016/j.jksuci.2023.101822.
25. Chen Sh., Wang T., Huang Zh., Hou X. Detection method of Golden Chip-Free Hardware Trojan based on the combination of ResNeXt structure and attention mechanism // Computers & Security. 2023. vol. 134. DOI: 10.1016/j.cose.2023.103428.
26. Rozesara M., Ghazinoori S., Manteghi M., Tabatabaeian S.H. A reverse engineering-based model for innovation process in complex product systems: Multiple case studies in the aviation industry // Journal of Engineering and Technology Management. 2023. vol. 69. DOI: 10.1016/j.jengtecman.2023.101765.
27. Lavanya T., Rajalakshmi K. Heterogenous ensemble learning driven multi-parametric assessment model for hardware Trojan detection // Integration. 2023. vol. 89. pp. 217–228. DOI: 10.1016/j.vlsi.2022.12.011.

28. Esirci F.N., Bayrakci A.A. Delay based hardware Trojan detection exploiting spatial correlations to suppress variations // Integration. 2023. vol. 91. pp. 107–118. DOI: 10.1016/j.vlsi.2023.03.006.
29. Cassano L., Iamundo M., Lopez T.A., Nazzari A., Di Natale G. DETON: DEfeating hardware Trojan horses in microprocessors through software ObfuscatioN // Journal of Systems Architecture. 2022. vol. 129. DOI: 10.1016/j.sysarc.2022.102592.
30. Damljanovic A., Ruospo A., Sanchez E., Squillero G. Machine learning for hardware security: Classifier-based identification of Trojans in pipelined microprocessors // Applied Soft Computing. 2022. vol. 116. DOI: 10.1016/j.asoc.2021.108068.
31. Palumbo A., Cassano L., Luzzi B., Hernández J.A., Reviriego P., Bianchi G., Ottavi M. Is your FPGA bitstream Hardware Trojan-free? Machine learning can provide an answer // Journal of Systems Architecture. 2022. vol. 128. DOI: 10.1016/j.sysarc.2022.102543.
32. Токарева Н.Н. Симметричная криптография. Краткий курс: учебное пособие / Под ред. А.Л. Перегожина // Новосибирск: Новосиб. гос. Ун-т., 2012. 234 с.
33. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы и исходный код на С. М.: Диалектика. 2017. 1040 с.
34. Макаренко С.И. Информационное противоборство и радиоэлектронная борьба в сетецентрических войнах начала XXI века. Санкт-Петербург: Наукоемкие технологии, 2017. 549 с. URL: http://psyfactor.org/t/Makarenko-InfPro_2017.pdf.
35. Elçi A., Pieprzyk J., Chefranov A.G., Orgun M.A., Wang H., Shankaran R. Theory and Practice of Cryptography Solutions for Secure Information Systems. Hershey, PA: IGI Global Scientific Publishing, 2013. 351 p. DOI: 10.4018/978-1-4666-4030-6.
36. Smart N.P. Cryptography Made Simple. Information Security and Cryptography series. Springer, 2016. 481 p. DOI: 10.1007/978-3-319-21936-3.
37. Бабенко Л.К., Ищукова Е.А. Современные алгоритмы блочного шифрования и методы их анализа. М.: Гелиос АРВ. 2006. 376 с.

Нагибин Дмитрий Владимирович — преподаватель, кафедра систем сбора и обработки информации, Федеральное государственное бюджетное военное образовательное учреждение высшего образования «Военно-космическая академия имени А.Ф. Можайского» Министерства обороны Российской Федерации. Область научных интересов: обратное проектирование аппаратных и программно-аппаратных комплексов, методы верификации сложных технических устройств, криптографические методы защиты информации. Число научных публикаций — 25. nagibin.86@internet.ru; улица Ждановская, 13, 197198, Санкт-Петербург, Россия; р.т.: +7(950)043-0162.

Петренко Алексей Сергеевич — инженер-исследователь, Федеральное государственное бюджетное военное образовательное учреждение высшего образования «Военно-космическая академия имени А.Ф. Можайского» Министерства обороны Российской Федерации. Область научных интересов: информационная безопасность, криптография, постквантовая криптография, квантовые вычисления, блокчейн, искусственный интеллект. Число научных публикаций — 280. a.petrenko1999@rambler.ru; улица Ждановская, 13, 197198, Санкт-Петербург, Россия; р.т.: +7(999)239-3901.

Давыденко Владислав Сергеевич — курсант, кафедра систем сбора и обработки информации, Федеральное государственное бюджетное военное образовательное учреждение высшего образования «Военно-космическая академия имени А.Ф. Можайского» Министерства обороны Российской Федерации. Область научных интересов: криптографические методы защиты информации, математические методы в криптографии. Число научных публикаций — 5. nagibin.86@internet.ru; улица Ждановская, 13, 197198, Санкт-Петербург, Россия; р.т.: +7(950)043-0162.

Котенко Игорь Витальевич — д-р техн. наук, профессор, заслуженный деятель науки Российской Федерации, главный научный сотрудник, руководитель лаборатории, лаборатория проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН). Область научных интересов: безопасность компьютерных сетей, в том числе управление политиками безопасности, разграничение прав доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер-терроризму. Число научных публикаций — 850. ivkote@comsec.spb.ru; 14-я линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)328-7181; факс: +7(812)328-4450.

Федорченко Елена Владимировна — канд. техн. наук, доцент, старший научный сотрудник, лаборатория проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН). Область научных интересов: безопасность информационных систем, методы анализа рисков компьютерных сетей, управление информационными рисками, анализ данных, поддержка принятия решений по повышению защищенности. Число научных публикаций — 137. doynikova@comsec.spb.ru; 14-я линия В.О., 39, 199178, Санкт-Петербург, Россия; р.т.: +7(812)328-7181.

Поддержка исследований. Работа выполнена при частичной финансовой поддержке бюджетной темы FFZF-2025-0016.

DOI 10.15622/ia.24.2.6

D. NAGIBIN, A. PETRENKO, V. DAVYDENKO, I. KOTENKO, E. FEDORCHENKO

INVASIVE APPROACH TO VERIFICATION OF FUNCTIONAL AND STRUCTURAL SPECIFICATIONS IMPLEMENTED IN CUSTOM INTEGRATED CIRCUITS

Nagibin D., Petrenko A., Davydenko V., Kotenko I., Fedorchenko E. **Invasive Approach to Verification of Functional and Structural Specifications Implemented in Custom Integrated Circuits.**

Abstract. An approach to verification of functional and structural specifications implemented in custom integrated circuits based on invasive research methods is presented. The relevance of this research is determined by the necessity of verification of functional-structural specifications supplied by third-party implementers of hardware implementations of information security algorithms, the difficulty of detecting modifications of these algorithms and undocumented capabilities implemented at the hardware level, and the lack of uniform, universal or standardized methods for solving this problem. The mathematical formulation of the research problem is specified; its essence is to verify the equality of the values of the declared specification parameters and their values restored by the reverse engineering method. The results of the application of the verification technique of functional and structural specifications are presented using examples of its adaptation to the study of hardware-implemented DES and AES encryption algorithms. The restored functional and structural blocks of the algorithms (in particular, the substitution block) were successfully verified.

Keywords: custom integrated circuit, identification, verification, functional and structural specifications, cryptographic algorithms.

References

1. Nagibin D.V., Platonov A.A., Sabirov T.R. [Technique for identification of functional-structural specifications realized in custom integrated circuits]. Trudy Voenno-kosmicheskoj akademii imeni A.F. Mozhajskogo – Proceedings of the Mozhaisky Military Space Academy. 2024. no. 690. pp. 112–120. (In Russ.).
2. Mustafa Dhiaa Al-Hassan, Qusay Zuhair Abdulla. Robust Password Encryption Technique with an Extra Security Layer. Iraqi Journal of Science. 2023. vol. 64. no. 3. pp. 1477–1486. DOI: 10.24996/ijs.2023.64.3.36.
3. Alsuwaiedi H.K.A., Rahma A.M.S. A new modified DES algorithm based on the development of binary encryption functions. Journal of King Saud University – Computer and Information Sciences. 2023. vol. 35(8). DOI: 10.1016/j.jksuci.2023.101716.
4. Agate V., Concone F., de Paola A., Ferraro P., Lo Re G., Morana M. Bayesian Modeling for Differential Cryptanalysis of Block Ciphers: A DES Instance. IEEE Access. 2023. vol. 11. pp. 4809–4820. DOI: 10.1109/ACCESS.2023.3236240.
5. Wu Y., Dai X. Encryption of accounting data using DES algorithm in Computing Environment. Journal of Intelligent & Fuzzy Systems. 2020. vol. 39. pp. 5085–5095. DOI: 10.3233/JIFS-179994.
6. Nitaj A., Susilo W., Tonien J. Enhanced S-boxes for the Advanced Encryption Standard with maximal periodicity and better avalanche property. Computer Standards & Interfaces. 2024. vol. 87. DOI: 10.1016/j.csi.2023.103769.
7. Zahid A.H., Arshad M.J. An innovative design of substitution-boxes using cubic polynomial mapping. Symmetry. 2019. vol. 11(3). DOI: 10.3390/sym11030437.

8. Zahid A.H., Arshad M.J., Ahmad M. Substitution-boxes using cubic fractional transformation. Entropy. 2019. vol. 21(3). DOI: 10.3390/e21030245.
9. Kotenko I.V., Levshun D.S., Chechulin A.A., Ushakov I.A., Krasov A.V. [Integrated approach to provide security of cyber-physical systems based on microcontrollers]. Voprosy kiberbezopasnosti – Cybersecurity issues. 2018. no. 3(27). pp. 29–38. DOI: 10.21681/2311-3456-2018-3-29-38. (In Russ.).
10. Desnitsky V., Chechulin A., Kotenko I., Levshun D., Kolomeec M. [Combined Design Technique for Secure Embedded Devices Exemplified by a Perimeter Protection System]. SPIIRAS Proceedings. 2016. no. 5(48). pp. 5–31. DOI: 10.15622/sp.48.1. (In Russ.).
11. Desnitsky V., Levshun D., Chechulin A., Kotenko I. Design technique for secure embedded devices: application for creation of integrated cyber-physical security system. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). 2016. vol. 7. pp. 60–80.
12. Kovalyov V., Kompaniets R., Novikov V [Verification of Programs Based on Similarity Relations]. SPIIRAS Proceedings. 2015. vol. 1(38). pp. 233–245. DOI: 10.15622/sp.38.13. (In Russ.).
13. Cousot P., Cousot R. A Galois connection calculus for abstract interpretation. Proceedings of the 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'14). ACM, 2014. pp. 3–4. DOI: 10.1145/2535838.2537850.
14. Fang Z., Darais D., Near J.P., Zhang Y. Zero Knowledge Static Program Analysis. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS'21). New York, USA: Association for Computing Machinery, 2021. pp. 2951–2967. DOI: 10.1145/3460120.3484795.
15. Kotenko I., Reznik S., Shorov A. [Security protocols verification combining existing approaches and tools]. SPIIRAS Proceedings. 2009. no. 8. pp. 292–310. DOI: 10.15622/sp.8.14. (In Russ.).
16. Tran D.D., Ogata K. IPSG: Invariant Proof Score Generator. 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). 2022. pp. 1050–1055. DOI: 10.1109/COMPSAC54236.2022.00164.
17. Riesco A., Ogata K., Futatsugi K. A Maude environment for cafeOBJ. Formal Aspects of Computing. 2017. vol. 29. pp. 309–334. DOI: 10.1007/s00165-016-0398-7.
18. Clavel M., Durán F., Eker S., Lincoln P., Martí-Oliet N., Meseguer J., Talcott C. All about Maude – a High-Performance Logical Framework. How to Specify, Program and Verify Systems in Rewriting Logic. Lecture Notes in Computer Science. 2007. 802 p. DOI: 10.1007/978-3-540-71999-1.
19. Tran D.D., Ogata K. Formal verification of TLS 1.2 by automatically generating proof scores. Computers & Security. 2022. vol. 123. DOI: 10.1016/j.cose.2022.102909.
20. Saeed Mahlaqa, Ibrar Muhammad, Mahmood Dua, Delshadi Amirmohammad, Saleemi Aqdas. Enhancing Computer Security through Formal Verification of Cryptographic Protocols Using Model Checking and Partial Order Techniques. The Asian Bulletin of Big Data Management. 2024. vol. 4(2). pp. 225–238. DOI: 10.62019/abbdm.v4i02.176.
21. Curaba C., D'Ambrosi D., Minisini A., Antol'in N.P.-C. CryptoFormalEval: Integrating LLMs and Formal Verification for Automated Cryptographic Protocol Vulnerability Detection. 2024. arXiv preprint arXiv:2411.13627.
22. Riesco A., Ogata K. An integrated tool set for verifying CafeOBJ specifications. Journal of Systems and Software. 2022. vol. 189. DOI: 10.1016/j.jss.2022.111302.
23. Awadhutkar P., Tamrawi A., Goluch R., Kothari S. Control flow equivalence method for establishing sanctity of compiling. Computers & Security. 2022. vol. 115. DOI: 10.1016/j.cose.2022.102608.

24. Cheng D., Dong Ch., He W., Chen Zh., Liu X., Zhang H. A fine-grained detection method for gate-level hardware Trojan based on bidirectional Graph Neural Networks. Journal of King Saud University – Computer and Information Sciences. 2023. vol. 35(10). DOI: 10.1016/j.jksuci.2023.101822.
25. Chen Sh., Wang T., Huang Zh., Hou X. Detection method of Golden Chip-Free Hardware Trojan based on the combination of ResNeXt structure and attention mechanism. Computers & Security. 2023. vol. 134. DOI: 10.1016/j.cose.2023.103428.
26. Rozesara M., Ghazinoori S., Manteghi M., Tabatabaeian S.H. A reverse engineering-based model for innovation process in complex product systems: Multiple case studies in the aviation industry. Journal of Engineering and Technology Management. 2023. vol. 69. DOI: 10.1016/j.jengtecman.2023.101765.
27. Lavanya T., Rajalakshmi K. Heterogenous ensemble learning driven multi-parametric assessment model for hardware Trojan detection. Integration. 2023. vol. 89. pp. 217–228. DOI: 10.1016/j.vlsi.2022.12.011.
28. Esirci F.N., Bayrakci A.A. Delay based hardware Trojan detection exploiting spatial correlations to suppress variations. Integration. 2023. vol. 91. pp. 107–118. DOI: 10.1016/j.vlsi.2023.03.006.
29. Cassano L., Iamundo M., Lopez T.A., Nazzari A., Di Natale G. DETON: DEfeating hardware Trojan horses in microprocessors through software ObfuscatioN. Journal of Systems Architecture. 2022. vol. 129. DOI: 10.1016/j.sysarc.2022.102592.
30. Damljanovic A., Ruospo A., Sanchez E., Squillero G. Machine learning for hardware security: Classifier-based identification of Trojans in pipelined microprocessors. Applied Soft Computing. 2022. vol. 116. DOI: 10.1016/j.asoc.2021.108068.
31. Palumbo A., Cassano L., Luzzi B., Hernández J.A., Reviriego P., Bianchi G., Ottavi M. Is your FPGA bitstream Hardware Trojan-free? Machine learning can provide an answer. Journal of Systems Architecture. 2022. vol. 128. DOI: 10.1016/j.sysarc.2022.102543.
32. Tokareva N.N. Simmetrichnaja kriptografija. Kratkij kurs: uchebnoe posobie [Simmetrichnaya kriptografiya (Symmetric cryptography). Short Course: Study Guide]. Novosibirsk: Novosib. State. Un-ty., 2012. 234 p. (In Russ.).
33. Schneier B. Protokoly, algoritmy i ishodnyj kod na S [Applied cryptography. Protocols, algorithms and source code in C Прикладная криптография]. M.: Dialektika. 2017. 1040 p. (In Russ.).
34. Makarenko S.I. Informacionnoe protivoborstvo i radiojelektronnaja bor'ba v setecentricheskih vojnah nachala XXI veka [Information warfare and electronic warfare in network-centric warfare of the early 21st century]. Sankt-Peterburg: Naukoemkie tehnologii, 2017. 549 p. (In Russ.).
35. Elçi A., Pieprzyk J., Chefranov A.G., Orgun M.A., Wang H., Shankaran R. Theory and Practice of Cryptography Solutions for Secure Information Systems. Hershey, PA: IGI Global Scientific Publishing, 2013. 351 p. DOI: 10.4018/978-1-4666-4030-6.
36. Smart N.P. Cryptography Made Simple. Information Security and Cryptography series. Springer, 2016. 481 p. DOI: 10.1007/978-3-319-21936-3.
37. Babenko L.K., Ishhukova E.A. Sovremennye algoritmy blochnogo shifrovanija i metody ih analiza [Modern block encryption algorithms and methods of their analysis.]. M.: Gelios ARV, 2006. 376 p. (In Russ.).

Nagibin Dmitry — Lecturer, Department of information collection and processing systems, Federal State Budgetary Military Educational Institution of Higher Education «Military Space Academy named after A.F. Mozhaisky» of the Ministry of Defense of the Russian Federation. Research interests: reverse engineering of hardware and hardware-software complexes, verification methods of complex technical devices, cryptographic methods of information

protection. The number of publications — 25. nagibin.86@internet.ru; 13, Zhdanovskaya St., 197198, St. Petersburg, Russia; office phone: +7(950)043-0162.

Petrenko Alexey — Research engineer, Federal State Budgetary Military Educational Institution of Higher Education «Military Space Academy named after A.F. Mozhaisky» of the Ministry of Defense of the Russian Federation. Research interests: information security, cryptography, post-quantum cryptography, quantum computing, blockchain, artificial intelligence. The number of publications — 280. a.petrenko1999@rambler.ru; 13, Zhdanovskaya St., 197198, St. Petersburg, Russia; office phone: +7(999)239-3901.

Davydenko Vladislav — Cadet, Department of information collection and processing systems, Federal State Budgetary Military Educational Institution of Higher Education «Military Space Academy named after A.F. Mozhaisky» of the Ministry of Defense of the Russian Federation. Research interests: cryptographic methods of information security, mathematical methods in cryptography. The number of publications — 5. nagibin.86@internet.ru; 13, Zhdanovskaya St., 197198, St. Petersburg, Russia; office phone: +7(950)043-0162.

Kotenko Igor — Ph.D., Dr.Sci., Professor, Honored scientist of the Russian Federation, Chief researcher, head of the laboratory, Laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: computer network security, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, malware protection, verification of security systems, digital right management, modeling, simulation and visualization technologies for counteraction to cyber terrorism. The number of publications — 850. ivkote@comsec.spb.ru; 39, 14-th Line V.O., 199178, St. Petersburg, Russia; office phone: +7(812)328-7181; fax: +7(812)328-4450.

Fedorchenko Elena — Ph.D., Associate Professor, Senior researcher, Laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS). Research interests: information systems security, risk analysis methods for computer networks, information security risk management, data analysis, security decision support. The number of publications — 137. doynikova@comsec.spb.ru; 39, 14-th Line V.O., 199178, St. Petersburg, Russia; office phone: +7(812)328-7181.

Acknowledgements. The reported study was partially funded by the budget project FFZF-2025-0016.