МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2022 **№ 56**

Зарегистрирован в Федеральной службе по надзору
в сфере связи и массовых коммуникаций

Свидетельство о регистрации ПИ № ФС 77-33762 от 16 октября 2008 г.

Подписной индекс в объединённом каталоге «Пресса России» 38696

УЧРЕДИТЕЛЬ
Томский государственный университет

Адрес редакции и издателя: 634050, г. Томск, пр. Ленина, 36
E-mail: pank@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и её приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании, теории надёжности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

Редактор *Н. И. Шидловская*
Верстка *И. А. Панкратовой*

Подписано к печати 30.05.2022. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 14,3. Тираж 300 экз.
Заказ № 5037. Цена свободная. Дата выхода в свет 03.06.2022.

Отпечатано на оборудовании
Издательства Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел.: 8(3822)53-15-28, 52-98-49

СОДЕРЖАНИЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 519.17 DOI 10.17223/20710410/56/1

ИССЛЕДОВАНИЕ ГРУППЫ АВТОМОРФИЗМОВ КОДА, АССОЦИИРОВАННОГО С ОПТИМАЛЬНОЙ КРИВОЙ РОДА ТРИ[1]

Е. С. Малыгина

Балтийский федеральный университет им. И. Канта, г. Калининград, Россия

E-mail: emalygina@kantiana.ru

Доказано, что отображение обладает свойством мультипликативности на соответствующем пространстве Римана — Роха, ассоциированного с дивизором mP_∞, который определяет некоторый алгебро-геометрический код (АГ-код), если число точек степени один функционального поля оптимальной кривой рода три, определённой над конечным полем с дискриминантом из $\{-19, -43, -67, -163\}$, имеет нижнюю границу $12m/(m-3)$. С помощью явного вычисления нормирования дивизоров полюсов образов базисных функций x, y, z функционального поля кривой при отображении λ установлено, что группа автоморфизмов функционального поля кривой является подгруппой автоморфизмов соответствующего АГ-кода. Доказано также, что при $m \geqslant 4$ и $n > 12m/(m-3)$ группа автоморфизмов функционального поля кривой изоморфна группе автоморфизмов АГ-кода, который ассоциирован с дивизорами $\sum\limits_{i=1}^{n} P_i$ и mP_∞, где P_i — точки степени один рассматриваемого функционального поля.

Ключевые слова: *оптимальная кривая, алгебро-геометрический код, функциональное поле, группа автоморфизмов кода.*

INVESTIGATION OF AUTOMORPHISM GROUP FOR CODE ASSOCIATED WITH OPTIMAL CURVE OF GENUS THREE

E. S. Malygina

Immanuel Kant Baltic Federal University, Kaliningrad, Russia

The main result of this paper is contained in two theorems. In the first theorem, it is proved that the mapping $\lambda : \mathscr{L}(mP_\infty) \to \mathscr{L}(mP_\infty)$ has the multiplicative property on the corresponding Riemann — Roch space associated with the divisor mP_∞ which defines some algebraic-geometric code if the number of points of degree one in the function field of genus three optimal curve over finite field with a discriminant $\{-19, -43, -67, -163\}$ has the lower bound $12m/(m-3)$. Using an explicit calculation with the valuations of the pole divisors of the images of the basis functions x, y, z in the function field of the curve via the mapping λ, we have proved that the automorphism

[1]Работа выполнена при финансовой поддержке Программы мобильности 5-100, а также при финансовой поддержке Минобрнауки России (соглашение №075-02-2022-872).

group of the function field of our curve is a subgroup in the automorphism group of the corresponding algebraic-geometric code. In the second theorem, it is proved that if $m \geqslant 4$ and $n > 12m/(m-3)$, then the automorphism group of the function field of our curve is isomorphic to the automorphism group of the algebraic-geometric code associated with divisors $\sum_{i=1}^{n} P_i$ and mP_∞, where P_i are points of the degree one.

Keywords: *optimal curve, algebraic-geometric code, function field, automorphism group of AG-code.*

Введение

Алгебро-геометрические коды (АГ-коды), также известные как коды Гоппы, впервые были представлены в [1]. В действительности, можно ассоциировать коды с дивизорами функциональных полей, тем самым рассматривая теорию кодирования с точки зрения алгебраической геометрии. Многие известные на сегодняшний день коды являются АГ-кодами или подмножествами АГ-кодов, что позволяет изучать их с алгебро-геометрической точки зрения. В этой работе мы рассмотрим приложения алгебраической геометрии исключительно с алгебраической точки зрения. Преимущество такого подхода позволяет взять за основу лишь некоторые базовые знания, касающиеся алгебраических расширений полей.

Улучшение границы Варшамова — Гилберта оказалось существенным прорывом в теории кодирования и позволило с большим интересом взглянуть на кривые с большим числом точек, а именно на их приложения. Знание группы автоморфизмов АГ-кода или даже её части примечательно тем, что позволяет получить информацию о структуре самого кода и зачастую может быть использовано в алгоритме декодирования.

В своё время Х. Штихтенот представил доказательство того, что при определённых условиях группа автоморфизмов рационального АГ-кода изоморфна некоторой подгруппе автоморфизмов соответствующего поля рациональных функций [2]. Чуть позже этот подход был обобщён Ч. Ксингом для специального класса эллиптических кодов [3] и эрмитовых кодов [4]. Вопрос подобного исследования для негиперэллиптических кривых рода $\geqslant 3$ на сегодняшний день остаётся открытым.

Данная работа посвящена исследованию группы автоморфизмов АГ-кодов, построенных на оптимальных кривых рода три, не являющихся гиперэллиптическими. Её суть заключается в установлении условия, связывающего число точек одного дивизора и кратность бесконечной точки второго дивизора. Оба этих дивизора ассоциированы с АГ-кодом, группа автоморфизмов которого изоморфна группе автоморфизмов соответствующего функционального поля оптимальной кривой.

В работе в качестве предварительных сведений представлены основная теоретическая база, касающаяся функциональных полей, и определение и свойства оптимальной кривой рода три. Ядром работы являются две теоремы: в первой определено условие, позволяющее сузить автоморфизмы функционального поля на пространство Римана — Роха, а далее рассмотреть их как автоморфизмы соответствующего АГ-кода; во второй теореме доказано наличие изоморфизма между группой автоморфизмов АГ-кода и группой автоморфизмов функционального поля кривой.

Предварительные результаты данной работы были представлены на конференции SIBECRYPT'18 [5].

1. Предварительные сведения

1.1. Общая теория функциональных полей

На протяжении всех предварительных сведений под K будем понимать произвольное поле, при необходимости внося уточнения.

Определение 1. Алгебраическим функциональным полем F/K от одной переменной называется расширение F поля K, такое, что F является конечным алгебраическим расширением $K(x)$ для некоторого элемента $x \in F$, являющегося трансцендентным над K.

Для краткости будем называть F/K просто функциональным полем. Будем считать известным понятие кольца нормирования функционального поля [6, Definition 1.1.4]. Здесь лишь упомянем о кольце нормирования точки P, подразумевая под P максимальный идеал кольца нормирования поля F/P:

$$\mathcal{O}_P = \{z \in F : z^{-1} \notin P\}.$$

При этом множество всех точек функционального поля F/K будем обозначать $\mathbb{P}_F$. Определим степень точки P функционального поля F/K как $\deg P = [\mathcal{O}_P/P : K]$.

Напомним, что, согласно [6], всякий элемент $0 \neq z \in F$ имеет единственное представление $z = t^n u$, если $P = t\mathcal{O}_P$, $u \in \mathcal{O}_P^*$ и $n \in \mathbb{Z}$.

Определение 2. С каждой точкой $P \in \mathbb{P}_F$ ассоциируем функцию

$$v_P : F \to \mathbb{Z} \cup \{\infty\},$$

которая играет роль дискретного нормирования функционального поля F/K:

$$v_P(z) = n \ \text{ для } \ z = t^n u, \quad v_P(0) = \infty.$$

Будем говорить, что точка P имеет нуль в z тогда и только тогда, когда $v_P(z) > 0$, и имеет полюс в z тогда и только тогда, когда $v_P(z) < 0$.

Отметим также, что дискретное нормирование v_P поля F/K удовлетворяет строгому неравенству треугольника

$$v_P(x + y) = \min\{v_P(x), v_P(y)\}, \tag{1}$$

если $v_P(x) \neq v_P(y)$ для $x, y \in F$.

Определение 3. Абелева группа $\mathcal{D}_F$, порождённая точками функционального поля F/K, называется группой дивизоров поля F/K.

Элементы группы $\mathcal{D}_F$ называются дивизорами поля F/K. Дивизор представляет собой формальную сумму точек:

$$D = \sum_{P \in \mathbb{P}_F} n_P P, \ \text{ где } \ n_P \in \mathbb{Z} \ \text{ и почти все } \ n_P = 0.$$

Носителем дивизора D является множество

$$\operatorname{supp}(D) = \{P \in \mathbb{P}_F : n_P \neq 0\}.$$

Для $P \in \mathbb{P}_F$ и дивизора D определим $v_P(D) = n_P$. Таким образом,

$$D = \sum_{P \in \operatorname{supp}(D)} v_P(D)\, P.$$

На $\mathcal{D}_F$ определено также частичное упорядочивание, а именно:

$$D_1 \leqslant D_2 \Leftrightarrow v_P(D_1) \leqslant v_P(D_2) \ \text{ для всех } \ P \in \mathbb{P}_F.$$

Степень дивизора определяется гомоморфизмом $\deg : \mathcal{D}_F \to \mathbb{Z}$:

$$\deg D = \sum_{P \in \mathbb{P}_F} v_P(D) \deg P.$$

Определение 4. Пусть $x \in F^*$. Обозначим через Z (через N) множество нулей (полюсов) x в $\mathbb{P}_F$. Тогда для функции x определим её дивизор нулей:

$$(x)_0 = \sum_{P \in Z} v_P(x)\, P;$$

дивизор полюсов:

$$(x)_\infty = \sum_{P \in N} (-v_P(x))\, P;$$

главный дивизор:

$$(x) = (x)_0 - (x)_\infty.$$

Дадим определение пространству Римана — Роха, одному из главных понятий в теории функциональных полей.

Определение 5. Для дивизора $D \in \mathcal{D}_F$ положим

$$\mathscr{L}(D) = \{x \in F : (x) \geqslant -D\} \cup \{0\}.$$

Данное множество называется пространством Римана — Роха, ассоциированным с дивизором D. Отметим, что $\mathscr{L}(D)$ является конечномерным векторным пространством над полем K. Целое $\dim D = \dim \mathscr{L}(D)$ называется размерностью дивизора D.

Определение 6. Род функционального поля F/K определён следующим образом:

$$g = \max\{\deg D - \dim D + 1 : D \in \mathcal{D}_F\}.$$

1.2. Алгебро-геометрические коды

Покажем, как задаётся код, ассоциированный с функциональным полем. Такие коды, как уже быо сказано, называются геометрическими кодами Гоппы или АГ-кодами.

Зафиксируем следующие обозначения:

— $F/\mathbb{F}_p$ — алгебраическое функциональное поле рода g;
— $P_1, P_2, \ldots, P_n$ — попарно различные точки поля $F/\mathbb{F}_p$ степени один;
— $D = P_1 + \ldots + P_n$;
— G — дивизор поля $F/\mathbb{F}_p$, такой, что $\operatorname{supp}(G) \cap \operatorname{supp}(D) = \varnothing$.

Определение 7. АГ-код $\mathcal{C}_{\mathscr{L}}(D, G)$, ассоциированный с дивизорами D и G, определён следующим образом:

$$\mathcal{C}_{\mathscr{L}}(D, G) = \{(x(P_1), \ldots, x(P_n)) : x \in \mathscr{L}(G)\} \subseteq \mathbb{F}_p^n.$$

Рассмотрим отображение

$$ev_D : \begin{cases} \mathscr{L}(G) \to \mathbb{F}_p^n, \\ ev_D(x) = (x(P_1), \ldots, x(P_n)) \in \mathbb{F}_p^n. \end{cases}$$

Это отображение $\mathbb{F}_p$-линейно, а код $\mathcal{C}_{\mathscr{L}}(D,G)$ является образом пространства Римана — Роха $\mathscr{L}(G)$ относительно этого отображеня. Если, кроме того, $\deg G < n$, то ev_D — биекция на $\mathcal{C}_{\mathscr{L}}(D,G)$.

Отметим, что, согласно [6, Theorem 2.2.2], $\mathcal{C}_{\mathscr{L}}(D,G)$ является $[n,k,d]$-кодом с параметрами $k = \dim G - \dim(G-D)$ и $d \geqslant n - \deg G$.

Прежде чем рассмотреть автоморфизмы АГ-кодов, рассмотрим автоморфизмы соответствующего функционального поля $F/\mathbb{F}_p$, которые образуют группу относительно композиции функций:

$$\mathrm{Aut}(F/\mathbb{F}_p) = \{\sigma : \sigma(a) = a,\ a \in \mathbb{F}_p\}.$$

Группа $\mathrm{Aut}(F/\mathbb{F}_p)$ действует на точки $\mathbb{P}_F$ как $\sigma(P) = \{\sigma(x) : x \in P\}$, при этом $\deg \sigma(P) = \deg P$. Действие $\mathrm{Aut}(F/\mathbb{F}_p)$ на $\mathbb{P}_F$ можно продолжить до действия на группу дивизоров, полагая

$$\sigma\left(\sum n_P P\right) = \sum n_P \sigma(P).$$

Под группой автоморфизмов произвольного кода $\mathcal{C} \subseteq \mathbb{F}_p^n$ будем понимать

$$\mathrm{Aut}(\mathcal{C}) = \{\pi \in S_n : \pi(\mathcal{C}) = \mathcal{C}\},$$

где S_n — симметрическая группа.

Определение 8. Определим

$$\mathrm{Aut}_{D,G}(F/\mathbb{F}_p) = \{\sigma \in \mathrm{Aut}(F/\mathbb{F}_q) : \sigma(D) = D, \sigma(G) = G\}.$$

Отметим, что автоморфизм $\sigma \in \mathrm{Aut}_{D,G}(F/\mathbb{F}_p)$ не обязательно фиксирует все точки $P_1, \ldots, P_n$, однако действует на них как перестановка. Окончательно получаем, что

$$\sigma(\mathscr{L}(G)) = \mathscr{L}(G),$$

поскольку $\sigma \in \mathrm{Aut}(F/\mathbb{F}_p)$ и $\sigma(G) = G$. Следующий результат показывает, что каждый автоморфизм $\sigma \in \mathrm{Aut}(F/\mathbb{F}_p)$ индуцирует автоморфизм соответствующего кода $\mathcal{C}_{\mathscr{L}}(D,G)$.

Теорема 1 [6, Proposition 8.2.3].

1) Действие группы $\mathrm{Aut}_{D,G}(F/\mathbb{F}_p)$ на код $\mathcal{C}_{\mathscr{L}}(D,G)$ задано следующим образом:

$$\sigma((x(P_1), \ldots, x(P_n))) = (x(\sigma(P_1)), \ldots, x(\sigma(P_n)))$$

для $x \in \mathscr{L}(G)$ и $\sigma \in \mathrm{Aut}_{D,G}(F/\mathbb{F}_p)$. Такое отображение является гомоморфизмом групп $\mathrm{Aut}_{D,G}(F/\mathbb{F}_p) \to \mathrm{Aut}(\mathcal{C}_{\mathscr{L}}(D,G))$.

2) Если $n > 2g+2$, где g — род поля $F/\mathbb{F}_p$, то этот гомоморфизм инъективен и, следовательно, группу автоморфизмов $\mathrm{Aut}_{D,G}(F/\mathbb{F}_p)$ можно рассматривать как подгруппу в $\mathrm{Aut}(\mathcal{C}_{\mathscr{L}}(D,G))$.

Согласно теореме 1, если $\deg D > 2g+2$, то $\mathrm{Aut}_{D,G}(F/\mathbb{F}_p)$ может быть вложена в $\mathrm{Aut}(\mathcal{C}_{\mathscr{L}}(D,G))$. Можно ассоциировать с каждым автоморфизмом π кода $\mathcal{C}_{\mathscr{L}}(D,G)$ линейный автоморфизм, обозначим его за λ_π, ассоциированного пространства Римана — Роха $\mathscr{L}(G)$. Автоморфизм λ_π является сужением на $\mathscr{L}(G)$ автоморфизма $\bar{\lambda} \in \mathrm{Aut}(F/\mathbb{F}_p)$ в том смысле, что $\bar{\lambda}|_{\mathscr{L}(G)}$ — линейное отображение и совпадает с λ_π на $\mathscr{L}(G)$ как $\mathbb{F}_p$-линейное отображение. Окончательно, $\bar{\lambda} \in \mathrm{Aut}_{D,G}(F/\mathbb{F}_p)$ и индуцирует π с помощью вложения $\mathrm{Aut}_{D,G}(F/\mathbb{F}_p) \hookrightarrow \mathrm{Aut}(\mathcal{C}_{\mathscr{L}}(D,G))$. В русле сказанного справедливо следующее

Определение 9. Пусть заданы два дивизора $D = P_1 + \ldots + P_n$ и $D' = P'_1 + \ldots + P'_n$, где $P_i, P'_i \in \mathbb{P}_{\mathbb{F}}$ — точки степени один. Пусть G — дивизор, такой, что $\operatorname{supp}(G) \cap \operatorname{supp}(D) = \varnothing$ и $\operatorname{supp}(G) \cap \operatorname{supp}(D') = \varnothing$, $\deg G < n$. Если $\mathcal{C}_{\mathscr{L}}(D, G) = \mathcal{C}_{\mathscr{L}}(D', G)$, то определим $\mathbb{F}_p$-линейное отображение $\lambda : \mathscr{L}(G) \to \mathscr{L}(G)$ следующим образом:

$$\begin{array}{ccc} \mathcal{C}_{\mathscr{L}}(D,G) & \xrightarrow{\mathsf{id}} & \mathcal{C}_{\mathscr{L}}(D',G) \\ \uparrow{\scriptstyle ev_D} & & \downarrow{\scriptstyle ev_{D'}^{-1}} \\ \mathscr{L}(G) & \xrightarrow[\lambda = ev_{D'}^{-1} \circ ev_D]{} & \mathscr{L}(G) \end{array}$$

Отметим, что корректность определения опирается на тот факт, что отображения ev_D и $ev_{D'}$ являются $\mathbb{F}_p$-биективными на $\mathcal{C}_{\mathscr{L}}(D, G)$.

Для всех $x \in \mathscr{L}(G)$ мы можем выразить $\lambda(x) \in \mathscr{L}(G)$ как $\lambda(x)(P'_i) = x(P_i)$ для $i = 1, \ldots, n$.

1.3. Определение и свойства оптимальной кривой

Определение 10. Пусть C — гладкая неприводимая проективная кривая рода g, определённая над конечным полем $\mathbb{F}_p$. Будем называть кривую C оптимальной, если число её рациональных точек удовлетворяет границе Хассе — Вейля — Серра

$$\#C(\mathbb{F}_p) = p + 1 \pm g\lfloor 2\sqrt{p} \rfloor.$$

В случае «$-$» кривая будет минимальной, в случае «$+$» — максимальной.

Согласно [7, Theorem 5.1], оптимальную кривую рода три можно задать явно. В работе рассмотрим лишь кривые, заданные уравнениями

$$\begin{cases} z^2 = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \beta_0 y, \\ y^2 = x^3 + ax + b, \end{cases} \tag{2}$$

где $\alpha_0, \alpha_1, \alpha_2, \beta_0, a, b \in \mathbb{F}_p$.

Под дискриминантом конечного поля $\mathbb{F}_p$ будем понимать число

$$d(\mathbb{F}_p) = \lfloor 2\sqrt{p} \rfloor^2 - 4p.$$

Замечание 1. Отметим, что, согласно [8], оптимальные кривые существуют также над конечными полями с дискриминантами $\{-43, -67, -163\}$ и могут быть заданы явно с помощью уравнения (2). Кроме того, такие кривые не являются гиперэллиптическими.

Теорема 2 [8, Предложение 3.1.5, с. 60]. Пусть C — оптимальная кривая рода три над конечным полем $\mathbb{F}_p$ с дискриминантом $d(\mathbb{F}_p) \in \{-19, -43, -67, -163\}$. Тогда

$$\operatorname{Aut}_{\mathbb{F}_p}(C) \cong D_3,$$

где D_3 — группа диэдра порядка 6.

1.4. Функциональное поле, ассоциированное с оптимальной кривой рода три

Рассмотрим функциональное поле $F/\mathbb{F}_p$. Существует несингулярная проективная кривая C (с точностью до изоморфизма), чьё функциональное поле $\mathbb{F}_p(C)$ изоморфно над $\mathbb{F}_p$ полю F. Кривая C может быть построена следующим образом: выбираются

$x, y, z \in F$, такие, что $F = \mathbb{F}_p(x, y, z)$. Пусть $f(X, Y, Z) \in \mathbb{F}_p[X, Y, Z]$ — неприводимый многочлен, такой, что $f(x, y, z) = 0$. Положим $W = \{P \in \mathbb{A}^3 : f(P) = 0\}$ и $\bar{W} \subseteq \mathbb{P}^3$ — проективное замыкание множества W. Полагая, что C — несингулярная модель $\bar{W}$, получаем, что $\mathbb{F}_p(C) \cong F$. Поэтому впредь будем отождествлять понятие кривой и её функционального поля.

Следующий результат достаточно элементарен, однако он понадобится для доказательства одной из основных теорем.

Теорема 3. Пусть дана башня числовых полей $K \subset K(x, y) \subset F = K(x, y, z)$, где x, y трансцендентны над K и $f(x, y, Z)$ — минимальный многочлен элемента z над $K(x, y)$. Предположим, что $\tilde{x}, \tilde{y}, \tilde{z} \in F \setminus K$ и пусть $f(\tilde{x}, \tilde{x}, \tilde{z}) = 0$. Тогда отображение

$$\lambda : \begin{cases} F/K \to F/K, \\ g(x, y, z) \mapsto g(\tilde{x}, \tilde{y}, \tilde{z}) \end{cases}$$

является K-эндоморфизмом поля F. Кроме того, если $x, y, z \in \operatorname{Im} \lambda$, то λ — автоморфизм расширения F/K.

Доказательство. Так как $\tilde{x}, \tilde{y}$ трансцендентны над K, то отображение

$$\lambda_1 : \begin{cases} K[x, y] \to K[\tilde{x}, \tilde{y}], \\ g(x, y) \mapsto g(\tilde{x}, \tilde{y}) \end{cases}$$

является изоморфизмом колец. Этот изоморфизм можно продолжить на поле дробей:

$$\lambda_1 \rightsquigarrow \lambda_2 : \begin{cases} K(x, y) \to K(\tilde{x}, \tilde{y}), \\ g(x, y) \mapsto g(\tilde{x}, \tilde{y}). \end{cases}$$

В этом случае λ_2 также является изоморфизмом и индуцирует изоморфизм колец:

$$\lambda_2 \rightsquigarrow \lambda_3 : \begin{cases} K(x, y)[Z] \to K(\tilde{x}, \tilde{y})[Z], \\ g(x, y, Z) \mapsto g(\tilde{x}, \tilde{y}, Z). \end{cases}$$

Окончательно получаем отображение λ_4:

$$\underbrace{K(x, y)[Z] \xrightarrow{\lambda_3} K(\tilde{x}, \tilde{y})[Z] \xrightarrow{\sigma} K(\tilde{x}, \tilde{y})[\tilde{z}]}_{\lambda_4 = \sigma \circ \lambda_3},$$

являющееся гомоморфизмом колец, где σ вычисляет значение рациональной функции от $\tilde{x}, \tilde{y}$ и переменной Z в точке $\tilde{z}$, т. е. $\sigma(g(\tilde{x}, \tilde{y}, Z)) = g(\tilde{x}, \tilde{y}, \tilde{z})$. Поскольку $\operatorname{Ker} \lambda_4 = (f(x, y, Z))$, то

$$F = K(x, y)[Z]/(f(x, y, Z)) \cong K(\tilde{x}, \tilde{y})[\tilde{z}] = K(\tilde{x}, \tilde{y})(\tilde{z}) \subseteq F.$$

Таким образом, отображение λ, индуцированное с помощью λ_4, является желаемым K-эндоморфизмом поля F. ■

Замечание 2. Рассмотрим функциональное поле $F/\mathbb{F}_p$ нашей оптимальной кривой. Согласно рассуждениям, представленным в [7], отметим, что

1) F не является рациональным;
2) существует точка $P_\infty \in \mathbb{P}_F$, $\deg P_\infty = 2$, и элементы $x, y, z \in F$, такие, что дивизоры полюсов этих функций имеют вид $(x)_\infty = (z)_\infty = 2P_\infty$, $(y)_\infty = 3P_\infty$;
3) для $m \geqslant 0$ элементы $x^i y^j z^k$ образуют базис пространства $\mathscr{L}(mP_\infty)$, где $i \geqslant 0$, $0 \leqslant j, k \leqslant 1$ и $2i + 3j + 2k \leqslant m$.

2. Основной результат

Исследуем класс алгебраических функциональных полей, которые не являются ни эллиптическими, ни гиперэллиптическими. Этот класс функциональных полей ассоциирован с оптимальными кривыми рода три. Покажем, при каких условиях группа автоморфизмов функционального поля совпадает с группой автоморфизмов соответствующего АГ-кода.

Зафиксируем некоторые обозначения:

— F — функциональное поле оптимальной кривой C;
— $J \subseteq \mathbb{P}_F \setminus \{P_\infty\}$;
— $n = |J|$;
— $D = \sum\limits_{P \in J} P = P_1 + P_2 + \ldots + P_n$;
— $D' = \sum\limits_{r=1}^{n} P_{\pi(r)}$ для некоторой перестановки $\pi \in S_n$.

В представленной далее теореме мы неоднократно будем ссылаться на следующий результат:

Лемма 1. Пусть $u, v \in \mathscr{L}(G)$ для некоторого дивизора $G = G_0 - G_1$, где $G_0, G_1 \geqslant 0$; $r = \deg G_0$; $P_1, P_2, \ldots, P_n$ — различные точки степени один, не лежащие в носителе дивизора G_0. Если $u(P_i) = v(P_i)$ для $i = 1, \ldots, n$ и $n > r$, то $u = v$.

Доказательство. Поскольку $\mathscr{L}(G)$ — векторное пространство, то $u - v \in \mathscr{L}(G)$. Следовательно, $\deg((u-v)_\infty) \leqslant r$. Однако функция $u - v$ имеет, как минимум, $n > r$ нулей, так как $(u-v)(P_i) = 0$ и $v_{P_i}(u-v) \geqslant 1$ для $i = 1, \ldots, n$. Принимая во внимание факт, что $\deg((u-v)_0) = \deg((u-v)_\infty)$ [6, Theorem 1.4.11], единственным вариантом является $u - v = 0$. ■

Нам потребуется установить взаимосвязь числа точек дивизора D с параметром m, где $G = mP_\infty$.

Теорема 4. Пусть $m \geqslant 4$ и $x, y, z \in \mathscr{L}(mP_\infty)$. Предположим, что $\mathcal{C}_{\mathscr{L}}(D, mP_\infty) = \mathcal{C}_{\mathscr{L}}(D', mP_\infty)$. Определим отображение $\lambda : \mathscr{L}(mP_\infty) \to \mathscr{L}(mP_\infty)$ так же, как в определении 9. Если выполняются следующие условия:

1) $-v_{P_\infty}(\lambda(x)) \leqslant c$, где $c \leqslant m$;
2) $n > 12m/(m-3)$,

то $v_{P_\infty}(\lambda(x)) = v_{P_\infty}(\lambda(z)) = -2$ и $v_{P_\infty}(\lambda(y)) = -3$. Кроме того, λ является сужением автоморфизма функционального поля кривой на $\mathscr{L}(mP_\infty)$.

Доказательство. Для $0 \leqslant j, k \leqslant 1$ определим $\alpha_{ij} = \lfloor (m - 3j - 2k)/2 \rfloor$. Тогда, очевидно, имеем

$$\begin{aligned}\mathscr{L}(mP_\infty) = \langle 1, x, x^2, \ldots, x^{\alpha_{00}};\\ y, xy, x^2y, \ldots, x^{\alpha_{10}}y;\\ z, xz, x^2z, \ldots, x^{\alpha_{01}}z;\\ yz, xyz, x^2yz, \ldots, x^{\alpha_{11}}yz\rangle.\end{aligned}$$

Докажем, что λ действует мультипликативно на $\mathscr{L}(mP_\infty)$. Другими словами, если $x^i y^j z^k \in \mathscr{L}(mP_\infty)$, то $\lambda(x^i y^j z^k) = \lambda(x)^i \lambda(y)^j \lambda(z)^k$ для $i \geqslant 0$, $0 \leqslant j, k \leqslant 1$. В ходе доказательства этого факта мы определим оценки степеней дивизоров полюсов функций $\lambda(x)$, $\lambda(y)$ и $\lambda(z)$.

Из условия теоремы 4 и того, что $(x)_\infty = (z)_\infty = 2P_\infty$, $(y)_\infty = 3P_\infty$, имеем

$$4 \leqslant \deg((\lambda(x)_\infty)) \leqslant c;$$
$$4 \leqslant \deg((\lambda(y)_\infty)), \deg((\lambda(z)_\infty)) \leqslant m.$$

Покажем, что $\lambda(x)$, $\lambda(y)$ и $\lambda(z)$ удовлетворяют уравнению (2) функционального поля кривой и, следовательно, определяют эндоморфизм $\tilde{\lambda}$ поля $F/\mathbb{F}_q$, такой, что $\tilde{\lambda} = \lambda|_{\mathscr{L}(mP_\infty)}$. Поскольку $x, y, z \in \operatorname{Im}\lambda$, то $x, y, z \in \operatorname{Im}\tilde{\lambda}$ и, следовательно, $\tilde{\lambda}$ — автоморфизм поля $F/\mathbb{F}_q$, доказывающий, что $v_{Q_\infty}(\lambda(x)) = v_{Q_\infty}(\lambda(z)) = -2$, $v_{Q_\infty}(\lambda(y)) = -3$.

Сначала покажем, что $\lambda(x^i) = \lambda(x)^i$ для $i = 1, \ldots, \alpha_{00}$. Выберем μ таким, что

$$\mu c \leqslant m < (\mu + 1)c \leqslant m + c.$$

Из условия теоремы $\deg((\lambda(x))_\infty) \leqslant c$ следует, что

$$\deg((\lambda(x)^2)_\infty) \leqslant 2c \leqslant m + c, \quad \text{поскольку} \quad c \leqslant m.$$

Тогда $\lambda(x^2)$, $\lambda(x)^2 \in \mathscr{L}((m + c)P_\infty)$, а по лемме 1 имеем

$$\lambda(x^2) = \lambda(x)^2 \quad \text{в} \quad n > m + c \quad \text{точках.}$$

Продолжая рассуждения, заключаем:

$$\lambda(x^i) = \lambda(x)^i \quad \text{для} \quad 1 \leqslant i \leqslant \mu.$$

Если $\mu = \alpha_{00}$, то получаем

$$-v_{P_\infty}(\lambda(x)) \leqslant \min\left\{c; \left[\frac{m}{\alpha_{00}}\right]\right\}.$$

Иначе считаем, что $\mu < \alpha_{00}$, и тогда $x^{\mu+1} \in \mathscr{L}(mP_\infty)$. Так как $\deg(\lambda(x)^{\mu+1})_\infty \leqslant m + c$, то получаем

$$\lambda(x^{\mu+1}) = \lambda(x)^{\mu+1},$$

следовательно, $m \geqslant -v_{P_\infty}(\lambda(x^{\mu+1})) = -(\mu + 1)\, v_{P_\infty}(\lambda(x))$. Таким образом,

$$-v_{P_\infty}(\lambda(x)) \leqslant \frac{m}{\mu + 1} < c.$$

Применяя лемму 1 к $\mathscr{L}((m + c)P_\infty)$ и учитывая, что $-v_{P_\infty}(\lambda(x)^{\mu+2}) \leqslant (\mu + 2)\,\dfrac{m}{\mu + 1} = m + \dfrac{m}{\mu + 1} < m + c < n$, получаем

$$\lambda(x^{\mu+2}) = \lambda(x)^{\mu+1} \cdot \lambda(x).$$

Поскольку $-v_{P_\infty}(\lambda(x^{\mu+2})) \leqslant m$, то

$$-v_{P_\infty}(\lambda(x)) \leqslant \frac{m}{\mu + 2}.$$

Продолжая этот процесс, находим

$$-v_{P_\infty}(\lambda(x)) \leqslant \frac{m}{\alpha_{00}}.$$

Окончательно заключаем, что в любом случае

$$4 \leqslant -v_{P_\infty}(\lambda(x)) \leqslant \min\left\{c; \left[\frac{m}{\alpha_{00}}\right]\right\}. \tag{3}$$

Отметим, что $\alpha_{00} = \left[\frac{m}{4}\right] \geqslant \frac{m-4+1}{4} = \frac{m-3}{4}$. Тогда $\left[\frac{m}{\alpha_{00}}\right] \leqslant \frac{4m}{m-3}$ и

$$-v_{P_\infty}(\lambda(x)) \leqslant \min\left\{c; \frac{4m}{m-3}\right\}. \tag{4}$$

Учитывая, что $n > m+c$ и $\lambda(yx)$, $\lambda(y)\lambda(x) \in \mathscr{L}((m+c)P_\infty)$, и снова применяя лемму 1, получаем

$$\lambda(yx) = \lambda(y)\lambda(x).$$

Переходя к нормированиям и учитывая, что изначально $\lambda(y) \in \mathscr{L}(mP_\infty)$, имеем $-v_{P_\infty}(\lambda(yx)) = -v_{P_\infty}(\lambda(y)) - v_{P_\infty}(\lambda(x)) \leqslant -v_{P_\infty}(\lambda(y))$. Следовательно,

$$-v_{P_\infty}(\lambda(y)) \leqslant m + v_{P_\infty}(\lambda(x)).$$

Рассматривая элемент $\lambda(yx^2)$, аналогичным образом заключаем, что $\lambda(yx^2) = \lambda(y)\lambda(x)^2$. Продолжая, получаем

$$\lambda(yx^s) = \lambda(y)\lambda(x)^s \quad \text{для} \quad s = 0, \ldots, \alpha_{10}.$$

Принимая во внимание, что элементы $y, xy, \ldots, x^{\alpha_{10}}y$ входят в систему образующих пространства $\mathscr{L}(mP_\infty)$, и рассуждая по аналогии с (3), получаем

$$-v_{P_\infty}(\lambda(y)) \leqslant m - 4\alpha_{10}.$$

Поскольку $\alpha_{10} = \left[\frac{m-3}{4}\right] \geqslant \frac{m-3-4+1}{4} = \frac{m-6}{4}$, то

$$-v_{P_\infty}(\lambda(y)) \leqslant 6. \tag{5}$$

Аналогично рассуждаем для оценки величины $-v_{P_\infty}(\lambda(z))$. Поскольку $n > m + c$ и $\lambda(zx), \lambda(z)\lambda(x) \in \mathscr{L}((m+c)P_\infty)$, то $\lambda(zx) = \lambda(z)\lambda(x)$. Имеем

$$-v_{P_\infty}(\lambda(z)) \leqslant m + v_{P_\infty}(\lambda(x)).$$

Принимая во внимание, что элементы $z, xz, \ldots, x^{\alpha_{01}}z$ входят в систему образующих пространства $\mathscr{L}(mP_\infty)$, получаем

$$-v_{P_\infty}(\lambda(z)) \leqslant m - 4\alpha_{01}.$$

Поскольку $\alpha_{01} = \left[\frac{m-2}{4}\right] \geqslant \frac{m-2-4+1}{4} = \frac{m-5}{4}$, то

$$-v_{P_\infty}(\lambda(z)) \leqslant 5. \tag{6}$$

Случай с элементами $yz, xyz, \ldots, x^{\alpha_{11}}yz$ мы не рассматриваем, поскольку органичились рассмотрением уравнения (2), в котором они не фигурируют.

Подействуем на определяющее уравнение λ, т. е. фактически осуществим замену x, y, z на $\lambda(x), \lambda(y), \lambda(z)$:

$$\begin{cases}(\lambda(z))^2(P_i') = (\alpha_0 + \alpha_1\lambda(x) + \alpha_2(\lambda(x))^2 + \beta_0\lambda(y))(P_i'),\\ (\lambda(y))^2(P_i') = ((\lambda(x))^3 + a\lambda(x) + b)(P_i').\end{cases}$$

В первом уравнении функция $\lambda(z)^2$ в левой части имеет не более 10 полюсов (в силу условия (6)). Учитывая (5), (4) и неравенство треугольника (1), функция в правой части $\alpha_0 + \alpha_1\lambda(x) + \alpha_2(\lambda(x))^2 + \beta_0\lambda(y)$ имеет не более $\min\{8m/(m-3); 6\}$ полюсов. Поскольку равенство $8m/(m-3) = 6$ невозможно из-за условия $m \geqslant 4$, то окончательно эта функция имеет не более $8m/(m-3)$ полюсов.

Во втором уравнении функция $\lambda(y)^2$ в левой части имеет не более 12 полюсов, а функция $(\lambda(x))^3 + a\lambda(x) + b$ — не более $12m/(m-3)$ полюсов (в силу условий (5) и (4)).

Очевидно, что $\max\{m+c, 10, 8m/(m-3), 12, 12m/(m-3)\} = 12m/(m-3)$ при $m \geqslant 4$. Поскольку по построению $n > 12m/(m-3)$, можем применить лемму 1. Таким образом, функции в обеих частях двух уравнений совпадают. Следовательно, λ сохраняет определяющее уравнение исходного функционального поля кривой, откуда по теореме 3 элементы $\lambda(x)$, $\lambda(y)$, $\lambda(z)$ определяют эндоморфизм $\tilde{\lambda}$, такой, что $\tilde{\lambda} = \lambda|_{\mathscr{L}(mP_\infty)}$. А так как $x, y, z \in \operatorname{Im}\tilde{\lambda}$, то $\tilde{\lambda}$ действует как автоморфизм соответственно $v_{P_\infty}(\lambda(x)) = v_{P_\infty}(\lambda(z)) = 2$ и $v_{P_\infty}(\lambda(y)) = 3$. ■

Используя теорему 4, можем определить группу автоморфизмов класса кодов, ассоциированных с функциональными полями оптимальных кривых рода три.

Теорема 5. Пусть $F/\mathbb{F}_p$ — функциональное поле оптимальной кривой рода три. Положим $m \geqslant 4$ и $D = \sum\limits_{i=1}^{n} P_i$, где $P_i \in J$, P_i — точки степени один и $J \subseteq \mathbb{P}_F \setminus \{P_\infty\}$. Если $n > 12m/(m-3)$, то

$$\operatorname{Aut}(\mathcal{C}_{\mathscr{L}}(D, mP_\infty)) \cong \operatorname{Aut}_{D,mP_\infty}(F/\mathbb{F}_p).$$

Доказательство. Согласно теореме 1, группа $\operatorname{Aut}_{D,mP_\infty}(F/\mathbb{F}_p)$ является подгруппой группы $\operatorname{Aut}(\mathcal{C}_{\mathscr{L}}(D, mP_\infty))$. Покажем, при каких условиях достигается изоморфизм этих групп.

Рассмотрим $\pi \in \operatorname{Aut}(\mathcal{C}_{\mathscr{L}}(D, mP_\infty))$. Поскольку $\mathcal{C}_{\mathscr{L}}(D, mP_\infty) = \mathcal{C}_{\mathscr{L}}(\pi(D), mP_\infty)$, в качестве π можем рассматривать отображение λ_π, ассоциированное с π и введённое в определении 9. Покажем, что λ_π является сужением автоморфизма $\tilde{\lambda} \in \in \operatorname{Aut}_{D,mP_\infty}(F/\mathbb{F}_p)$.

По теореме 4 автоморфизм λ_π можно рассматривать как автоморфизм $\tilde{\lambda}$ функционального поля F на $\mathscr{L}(mP_\infty)$. Так как $\dim(mP_\infty) > 1$, то $\mathscr{L}(mP_\infty) \setminus \mathbb{F}_p \neq \varnothing$. Тогда существует функция $h \in \mathscr{L}(mP_\infty)$, такая, что h имеет полюс лишь в точке P_∞ и ни в какой другой точке. Имеем $\mathscr{L}(mP_\infty) = \tilde{\lambda}(\mathscr{L}(mP_\infty)) = \mathscr{L}(\tilde{\lambda}(mP_\infty))$. Кроме того, $\tilde{\lambda}(mP_\infty) = m\tilde{\lambda}(P_\infty)$ и, таким образом, $\tilde{\lambda}(P_\infty) = P_\infty$, иначе функция h будет иметь полюс в точке $\tilde{\lambda}(P)$, что невозможно по предположению. Следовательно, $\tilde{\lambda}(mP_\infty) = mP_\infty$.

Имеем $\mathcal{C}_{\mathscr{L}}(\pi(D), mP_\infty) = \mathcal{C}_{\mathscr{L}}(D, mP_\infty) = \mathcal{C}_{\mathscr{L}}(\tilde{\lambda}(D), \tilde{\lambda}(mP_\infty)) = \mathcal{C}_{\mathscr{L}}(\tilde{\lambda}(D), mP_\infty)$ и $\tilde{\lambda}(h) = \lambda(h)$ для всех $h \in \mathscr{L}(mP_\infty)$. Следовательно, $\tilde{\lambda}(h)(\tilde{\lambda}(P_i)) = \lambda(h)(\pi(P_i))$ для всех $h \in \mathscr{L}(mP_\infty)$ и $i = 1, \ldots, n$.

Поскольку $F = \mathbb{F}_p(x, y, z)$, то если $P \in \mathbb{P}_F$ — точка степени один, значения $x(P)$, $y(P)$ и $z(P)$ единственным образом определены в точке P. Так как $x, y, z \in \mathscr{L}(mP_\infty)$, то $x(\tilde{\lambda}(P_i)) = x(\pi(P_i))$, $y(\tilde{\lambda}(P_i)) = y(\pi(P_i))$, $z(\tilde{\lambda}(P_i)) = z(\pi(P_i))$ и, следовательно, $\tilde{\lambda}(P_i) = \pi(P_i)$ для $i = 1, \ldots, n$. Получаем $\tilde{\lambda}(D) = \pi(D)$, откуда $\tilde{\lambda} \in \operatorname{Aut}_{D,mP_\infty}(F/\mathbb{F}_p)$. ■

Приведём пример, ориентируясь на некоторые данные из [8].

Пример 1. Рассмотрим максимальную кривую, определённую над конечным полем $\mathbb{F}_{47}$:

$$\begin{cases} z^2 = 5 + 45x + 30x^2 + 10y, \\ y^2 = x^3 + x + 38. \end{cases}$$

Число её $\mathbb{F}_{47}$-рациональных точек, удовлетворяющих границе Хассе — Вейля — Серра, равно 87. Отметим, что рациональные точки кривой соответствуют точкам степени один её функционального поля $F/\mathbb{F}_{47}$, поэтому в качестве дивизора D мы можем рассмотреть дивизор $D = P_1 + \ldots + P_{87}$.

Для $m = 4$ рассмотрим дивизор $G = 4P_\infty$. Тогда, применяя теорему 5, получаем

$$\mathrm{Aut}(\mathcal{C}_{\mathscr{L}}(D, 4P_\infty)) \cong \mathrm{Aut}_{D,4P_\infty}(F/\mathbb{F}_p) \cong \mathrm{Aut}_{\mathbb{F}_{47}}(C) \cong D_3.$$

ЛИТЕРАТУРА

1. *Goppa V. D.* Geometry and Codes. Kluwer Academic Publishers, 1988.
2. *Stichtenoth H.* On automorphisms of geometric Goppa codes // J. Algebra. 1990. V. 130. P. 113–121.
3. *Xing C.* Automorphism group of elliptic codes // Comm. Algebra. 1995. V. 23. No. 11. P. 4061–4072.
4. *Xing C.* On automorphism groups of the Hermitian codes // IEEE Trans. Inform. Theory. 1995. V. 41. No. 6. P. 1629–1635.
5. *Малыгина Е. С.* Исследование группы автоморфизмов кода, ассоциированного с оптимальной кривой рода три // Прикладная дискретная математика. Приложение. 2018. № 11. С. 115–117.
6. *Stichtenoth H.* Algebraic Function Fields and Codes. Springer Verlag, 1991.
7. *Alekseenko E., Aleshnikov S., Markin N., and Zaytsev A.* Optimal curves over finite fields with discriminant -19 // Finite Fields and Their Appl. 2011. V. 17. Iss. 4. P. 350–358.
8. *Alekseenko E.* Explicit construction of optimal curves of genus three. PhD Thesis. 2016. `http://iitp.ru/upload/content/1203/AES_disser.pdf`

REFERENCES

1. *Goppa V. D.* Geometry and Codes. Kluwer Academic Publishers, 1988.
2. *Stichtenoth H.* On automorphisms of geometric Goppa codes. J. Algebra, 1990, vol. 130, pp. 113–121.
3. *Xing C.* Automorphism group of elliptic codes. Comm. Algebra, 1995, vol. 23, no. 11, pp. 4061–4072.
4. *Xing C.* On automorphism groups of the Hermitian codes. IEEE Trans. Inform. Theory, 1995, vol. 41, no. 6, pp. 1629–1635.
5. *Malygina E. S.* Issledovanie gruppy avtomorfizmov koda, assotsiirovannogo s optimal'noy krivoy roda tri [Investigation of automorphism group for code associated with optimal curve of genus three]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2018, no. 11, pp. 115–117. (in Russian)
6. *Stichtenoth H.* Algebraic Function Fields and Codes. Springer Verlag, 1991.
7. *Alekseenko E., Aleshnikov S., Markin N., and Zaytsev A.* Optimal curves over finite fields with discriminant -19. Finite Fields and Their Appl., 2011, vol. 17, iss. 4, pp. 350–358.
8. *Alekseenko E.* Explicit Construction of Optimal Curves of Genus Three. PhD Thesis, 2016. `http://iitp.ru/upload/content/1203/AES_disser.pdf`

УДК 512.643 DOI 10.17223/20710410/56/2

THE CONSTRUCTION OF CIRCULANT MATRICES RELATED TO MDS MATRICES

S. S. Malakhov, M. I. Rozhkov

HSE University, Moscow, Russia

E-mail: ssmalakhov@edu.hse.ru, mirozhkov@hse.ru

The objective of this paper is to suggest a method of the construction of circulant matrices, which are appropriate for being MDS (Maximum Distance Separable) matrices utilising in cryptography. Thus, we focus on designing so-called bi-regular circulant matrices, and furthermore, impose additional restraints on matrices in order that they have the maximal number of some element occurrences and the minimal number of distinct elements. The reason to construct bi-regular matrices is that any MDS matrix is necessarily the bi-regular one, and two additional restraints on matrix elements grant that matrix-vector multiplication for the samples constructed may be performed efficiently. The results obtained include an upper bound on the number of some element occurrences for which the circulant matrix is bi-regular. Furthermore, necessary and sufficient conditions for the circulant matrix bi-regularity are derived. On the basis of these conditions, we developed an efficient bi-regularity verification procedure. Additionally, several bi-regular circulant matrix layouts of order up to 31 with the maximal number of some element occurrences are listed. In particular, it appeared that there are no layouts of order 32 with more than 5 occurrences of any element which yield a bi-regular matrix (and hence an MDS matrix).

Keywords: *circulant matrix, MDS code, MDS matrix.*

О ПОСТРОЕНИИ ЦИРКУЛЯНТНЫХ МАТРИЦ, СВЯЗАННЫХ С MDS-МАТРИЦАМИ

С. С. Малахов, М. И. Рожков

Национальный исследовательский университет «Высшая школа экономики», г. Москва, Россия

Цель данной работы — предложить метод построения таких циркулянтных матриц, которые могут быть MDS-матрицами, используемыми в криптографии. Мы рассматриваем так называемые би-регулярные циркулянтные матрицы и, кроме того, налагаем на них дополнительные ограничения с тем, чтобы они имели максимальное число вхождений некоторого элемента и минимальное количество различных элементов. Интерес к би-регулярным матрицам обусловлен тем, что любая MDS-матрица обязательно является би-регулярной, а дополнительные ограничения на элементы матриц позволяют эффективнее реализовывать матрично-векторные операции с использованием таких матриц. Полученные результаты включают верхнюю границу числа вхождений некоторого элемента, при котором циркулянтная матрица остаётся би-регулярной, а также необходимые и достаточные условия би-регулярности циркулянтной матрицы. Кроме того, описан эффективный алгоритм проверки би-регулярности циркулянтной матрицы.

С его помощью построены шаблоны би-регулярных циркулянтных матриц порядка до 31 с максимальным числом вхождений некоторого элемента и установлено отсутствие би-регулярных циркулянтных матриц (и следовательно, MDS-матриц) порядка 32 с более чем пятью вхождениями одного элемента.

Ключевые слова: *циркулянтная матрица, МДР-код, MDS-код, MDS-матрица.*

1. Introduction

Suppose that $\boldsymbol{M}$ is a $k \times m$ matrix over a finite field $\mathbb{F}_q$. Then a set

$$\left\{(\boldsymbol{x}, \boldsymbol{x} \cdot \boldsymbol{M}) : \boldsymbol{x} \in (\mathbb{F}_q)^k\right\}$$

is a linear $[n, k, d]$ code of the length $n = k + m$ and the dimension k with the minimum Hamming distance d between any two code words. For a linear $[n, k, d]$ code the Singleton bound holds [1]:

$$d \leqslant n - k + 1 = m + 1.$$

A code with $d = m + 1$ is called the MDS code (Maximum Distance Separable code), and the corresponding matrix $\boldsymbol{M}$ is referred to as the MDS matrix.

The problem of MDS code existence relates to Segre's MDS conjecture proposed in [2]. It suggests that a set S of vectors of the vector space $(\mathbb{F}_q)^k$ such that every subset of S of size $k \leqslant q$ is a basis, comprises at most $q + 1$ elements, unless q is even and $k = 3$ or $k = q - 1$, in which case it comprises at most $q + 2$ elements. S. Ball has shown in [3] that S generates an MDS code and proved that a linear MDS code with the dimension $k \leqslant q$ has the length at most $q + k + 1 - \min\{k, \operatorname{char} \mathbb{F}_q\}$.

Furthermore, it is shown in [1, p. 321] that a linear code is MDS if and only if every square submatrix of $\boldsymbol{M}$ is non-singular. Therefore, we will define the MDS matrix as follows.

Definition 1. A matrix $\boldsymbol{M}$ is the MDS matrix if every square submatrix of $\boldsymbol{M}$ is non-singular.

MDS matrices are demanded for block cryptographic algorithms, where they are responsible for the input diffusion. An MDS matrix performs a linear transformation of an input block $\boldsymbol{x}$ of the following property: if $i, 1 \leqslant i \leqslant k$, elements of $\boldsymbol{x}$ are altered, then at least $m - i + 1$ elements of the output block $\boldsymbol{x} \cdot \boldsymbol{M}$ alter, where both the input and the output blocks can be interpreted as vectors of a k-dimensional vector space over a finite field $\mathbb{F}_q$. In this sense, MDS matrices provide perfect diffusion [4]. Several algorithms utilize MDS matrices including block ciphers Rijndael, GOST R 34.12-2015, IDEA NXT and hash functions GOST R 34.11-2012 and Whirlpool.

Although construction of MDS matrices is a computationally hard problem in general case, there are plenty of different particular techniques. One approach presumes that a specific matrix layout comprising variables is set. Then, variables are initialized with concrete values, and the resulting matrix is tested for being the MDS matrix. The approach described was proposed in [4] and performed in [5]. Not every matrix layout may produce MDS matrices, and therefore, it is of an interest to filter those layouts which never produce any. A method to filter matrix layouts is to verify their bi-regularity. The definition of the bi-regularity is given below.

Definition 2. Let $\mathscr{K}$ be a subset of a multiplicative group. The 2×2 matrix over $\mathscr{K}$ is bi-regular if at least in one row and one column there are two distinct entries. An arbitrary $k \times m$ matrix over $\mathscr{K}$ is bi-regular if every its 2×2 submatrix is bi-regular.

Remark 1. One may distinguish two particular cases, when $\mathscr{K}$ is exactly a multiplicative subgroup of a finite field, and when $\mathscr{K}$ represents a set of variables that cannot take nought values.

It is obvious that an MDS matrix is necessarily bi-regular, and so is a matrix layout that produces MDS matrices.

This paper focuses on the construction of bi-regular circulant matrix layouts which yield bi-regular matrices and hence may produce MDS matrices.

Definition 3. A circulant matrix denoted by its zeroth row $(a_0, \ldots, a_{m-1})$ is a matrix of the form

$$\begin{pmatrix} a_0 & a_1 & \cdots & a_{m-2} & a_{m-1} \\ a_1 & a_2 & \cdots & a_{m-1} & a_0 \\ \cdots & & & & \\ a_{m-2} & a_{m-1} & a_0 & \cdots & a_{m-3} \\ a_{m-1} & a_0 & \cdots & a_{m-3} & a_{m-2} \end{pmatrix}.$$

Remark 2. Speaking more generally, as rows of a matrix may be circularly shifted to the left or to the right, there exist two types of circulant matrices. Although this paper takes a left shift case as a basis for description, all the techniques presented are essentially applicable to both circulant matrix types.

Previously, circulant matrices were studied in several papers, for instance, in [6 – 9]. It was proved in [6] that circulant MDS matrices over a finite field of characteristic 2 are neither involutary nor orthogonal. However, [7] reveals that involutory circulant MDS matrices over the ring of matrices whose entries lie in characteristic 2 field do exist. The authors of [9] managed to construct circulant MDS matrices over the general linear group over the two-element field, and in [8] the authors studied circulant-like MDS matrices.

The objective of this paper is to suggest a method for the construction of bi-regular circulant matrices with the maximal number of some element occurrences and the minimal number of distinct elements. These two additional restraints on matrix elements allow performing matrix-vector multiplication more efficiently. The results obtained include the upper bound of the number of some element occurrences for which the circulant matrix bi-regularity preserves. Furthermore, necessary and sufficient conditions for the circulant matrix bi-regularity are derived. On the basis of these conditions, we developed an efficient bi-regularity verification procedure. Additionally, several bi-regular circulant matrix layouts of order up to 31 with the maximal number of some element occurrences are listed. In particular, it appeared that there are no layouts of order 32 with more than 5 occurrences of any element which yield a bi-regular matrix (and hence an MDS matrix).

This paper follows the report *On the construction of bi-regular circulant matrices, relating to MDS matrices* [10] made at the conference *Engineering Technologies and Informatics: Innovations and Applications* (En&T-2021).

The paper consists of two parts, not taking the introduction and the conclusion into account. The first part carries proofs for the upper bound of the number of arbitrary element occurrences together with the proof of necessary and sufficient conditions for the circulant matrix be-regularity. The second part contains instances of bi-regular circulant matrix layouts of an order up to 31 with maximum number of a given element occurrences.

2. Necessary and sufficient conditions for the circulant matrix bi-regularity

The following Lemma 1 provides one of the necessary conditions for the circulant matrix bi-regularity.

Lemma 1. Let $\boldsymbol{M}$ be a bi-regular circulant matrix over a subset $\mathscr{K}$ of some multiplicative group, denoted by its zeroth row $(a_0, \ldots, a_{m-1})$. Suppose that an element α is in the positions with indices $i_0, \ldots, i_{t-1}$, $t > 1$. Then the set of differences between two distinct indices

$$D_\alpha = \{(i - i') \bmod m : i \in \{i_0, \ldots, i_{t-1}\} \ni i', i \neq i'\}$$

comprises $t\,(t-1)$ elements.

Proof. Suppose that there exist indices $i_r < i_s$ and $i_u < i_v$ of the positions occupied by an element α such that $i_s - i_r = i_v - i_u$ or $i_s - i_r = m - i_v + i_u$. The following three cases are possible.

I. If $i_s - i_r = i_v - i_u$, while $i_r < i_s \leqslant i_u < i_v$, then in the zeroth row and in the row obtained from it by the $(i_u - i_r)$-position left circular shift there is an element α in the columns i_r and i_s. Hence, $\boldsymbol{M}$ is not bi-regular:

$$\begin{array}{c} \\ 0 \\ \\ i_u - i_r \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & i_r & & i_s & & i_u & & i_v & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \alpha & \cdots & \alpha & \cdots & \alpha & \cdots \\ & \vdots & & & & & & & \\ \cdots & \alpha & \cdots & \alpha & \cdots & & & & \end{pmatrix} \end{array}.$$

II. If $i_s - i_r = i_v - i_u$, while $i_r < i_u < i_s < i_v$, then in the zeroth row and in the row obtained from it by the $(i_u - i_r)$-position left circular shift there is an element α in the columns i_r and i_s. Hence, $\boldsymbol{M}$ is not bi-regular:

$$\begin{array}{c} \\ 0 \\ \\ i_u - i_r \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & i_r & & i_u & & i_s & & i_v & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \alpha & \cdots & \alpha & \cdots & \alpha & \cdots \\ & \vdots & & & & & & & \\ \cdots & \alpha & \cdots & & & \alpha & \cdots & & \end{pmatrix} \end{array}.$$

III. If $i_s - i_r = (i_u - i_v) \bmod m$, while $i_r \leqslant i_u < i_v \leqslant i_s$, then in the zeroth row and in the row obtained from it by the $(i_v - i_r)$-positions left circular shift there is an element α in the columns i_r and i_s. Hence, $\boldsymbol{M}$ is not bi-regular:

$$\begin{array}{c} \\ 0 \\ \\ i_v - i_r \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & i_r & & i_u & & i_v & & i_s & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \alpha & \cdots & \alpha & \cdots & \alpha & \cdots \\ & \vdots & & & & & & & \\ \cdots & \alpha & \cdots & & & & & \alpha & \cdots \end{pmatrix} \end{array}.$$

Thus, all the cases possible contradict to the matrix $\boldsymbol{M}$ bi-regularity. ■

Now we derive the upper bound of the number of arbitrary element occurrences in the bi-regular circulant matrix. Note that K. Zarankiewicz in [11] addressed the problem equivalent to finding the largest positive integer $z(k, m, p, q)$ such that a binary $k \times m$ matrix containing $z(k, m, p, q)$ ones may not have a $p \times q$ submatrix consisting entirely of ones. If we now take $p = q = 2$, then Zarankiewicz's problem is to find the largest number of arbitrary element occurrences at which the matrix bi-regularity preserves. I. Reiman proved in [12] that

$$z(k, m, 2, 2) \leqslant 1/2 \left(k + \left(k^2 + 4km(m-1)\right)^{1/2}\right),$$

$$z(t^2 - t + 1, t^2 - t + 1, 2, 2) = t^3 - t^2 + t.$$

It is an immediate corollary to Lemma 1 that for a circulant matrix of order $m = t(t-1)+1$ the maximal number of element occurrences, at which the matrix still can be bi-regular, meets the upper bound proved by Reiman, i.e., $z(m, m, 2, 2) = t^3 - t^2 + t$. Besides, the next corollary shows that Reiman's inequality remains strong enough for circulant matrices.

Corollary 1. Under conditions of the Lemma 1, the following inequality holds:

$$m \leqslant mt \leqslant 1/2 \left(m + \left(m^2 + 4m^2(m-1)\right)^{1/2}\right).$$

Proof. On the one hand, Lemma 1 asserts that the set D_α of differences of two distinct indices comprises $t\,(t-1)$ elements. On the other hand, the aggregate number of differences between two distinct indices does not exceed $m-1$. Therefore,

$$t\,(t-1) \leqslant m-1,$$

and hence,

$$1 \leqslant t \leqslant 1/2 + (m - 3/4)^{1/2} \Leftrightarrow m \leqslant mt \leqslant 1/2 \left(m + \left(m^2 + 4m^2(m-1)\right)^{1/2}\right).$$

The Corollary 1 is proved. ■

Remark 3. It is noteworthy that D_α is a difference set in case $t\,(t-1) = m-1$.

The next Lemma introduces an interrelationship between numbers of different element occurrences in a bi-regular matrix.

Lemma 2. Let $\boldsymbol{M}$ be a bi-regular circulant matrix over a subset $\mathscr{K}$ of some multiplicative group denoted by its zeroth row $(a_0, \ldots, a_{m-1})$. Suppose that an element α is in the positions with the indices $i_0, \ldots, i_{t_\alpha - 1}$, $t_\alpha > 1$, and an element β is in the positions with the indices $j_0, \ldots, j_{t_\beta - 1}$, $t_\beta > 1$. Then the sets of differences between two distinct indices of α and β

$$D_\alpha = \left\{(i - i')_{\bmod m} : i \in \{i_0, \ldots, i_{t_\alpha - 1}\} \ni i', i \neq i'\right\}$$
$$D_\beta = \left\{(j - j')_{\bmod m} : j \in \{j_0, \ldots, j_{t_\beta - 1}\} \ni j', j \neq j'\right\}$$

are disjoint.

Proof. Suppose that there exist indices $i_r < i_s$ and $j_u < j_v$ of positions occupied by an element α and an element β respectively. The following three cases are possible.

I. If $i_s - i_r = j_v - j_u$, while $i_r < i_s < j_u < j_v$, then in the columns i_r and i_s there are the element α in the zeroth row and the element β in the row obtained from the zeroth one by the $(j_u - i_r)$-position left circular shift. Hence, $\boldsymbol{M}$ is not bi-regular:

$$\begin{array}{c} \\ 0 \\ \\ j_u - i_r \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & i_r & & i_s & & j_u & & j_v & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \alpha & \cdots & \beta & \cdots & \beta & \cdots \\ & \vdots & & & & & & & \\ \cdots & \beta & \cdots & \beta & \cdots & & & & \end{pmatrix} \end{array}.$$

II. If $i_s - i_r = j_v - j_u$, while $i_r < j_u < i_s < j_v$, then in the columns i_r and i_s there are the element α in the zeroth row and the element β in the row obtained from the zeroth one by the $(j_u - i_r)$-positions left circular shift. Hence, $\boldsymbol{M}$ is not bi-regular:

$$\begin{array}{c} \\ 0 \\ \\ j_u - i_r \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & i_r & & j_u & & i_s & & j_v & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \beta & \cdots & \alpha & \cdots & \beta & \cdots \\ & \vdots & & & & & & & \\ \cdots & \beta & \cdots & & & \beta & \cdots & & \end{pmatrix} \end{array}.$$

III. If $i_s - i_r = m - (j_v - j_u)$, while $i_r < j_u < j_v < i_s$, then in the columns i_r and i_s there are the element α in the zeroth row and the element β in the row obtained from the zeroth one by the $(j_v - i_r)$-positions left circular shift. Hence, $\boldsymbol{M}$ is not bi-regular:

$$\begin{array}{c} \\ 0 \\ \\ j_v - i_r \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & i_r & & j_u & & j_v & & i_s & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \beta & \cdots & \beta & \cdots & \alpha & \cdots \\ & \vdots & & & & & & & \\ \cdots & \beta & \cdots & & & & & \beta & \cdots \end{pmatrix} \end{array}.$$

Thus, all the cases possible contradict to the matrix $\boldsymbol{M}$ bi-regularity. ■

The following Theorem provides the necessary and sufficient conditions for the circulant matrix bi-regularity.

Theorem 1. Let $\boldsymbol{M}$ be an $m \times m$ circulant matrix over a subset $\mathscr{K}$ of some multiplicative group, denoted by its zeroth row $(a_0, \ldots, a_{m-1})$. Suppose that an element α is in the positions with the indices $i_0, \ldots, i_{t_\alpha - 1}$, $t_\alpha > 1$, and an element β is in the positions with the indices $j_0, \ldots, j_{t_\beta - 1}$, $t_\beta > 1$. Let

$$D_\alpha = \left\{ (i - i') \bmod m : i \in \{i_0, \ldots, i_{t_\alpha - 1}\} \ni i', i \neq i' \right\}$$
$$D_\beta = \left\{ (j - j') \bmod m : j \in \{j_0, \ldots, j_{t_\beta - 1}\} \ni j', j \neq j' \right\}$$

be the sets of differences between two distinct indices of the positions occupied by α and β respectively. Then the matrix $\boldsymbol{M}$ is bi-regular if and only if for each such α and β:

1) the set D_α comprises $t_\alpha (t_\alpha - 1)$ elements, while D_β comprises $t_\beta (t_\beta - 1)$ elements;
2) the sets D_α and D_β are disjoint.

Proof. The necessity immediately follows from Lemmas 1 and 2.

To prove sufficiency, suppose the matrix $\boldsymbol{M}$ is not bi-regular. The following three cases are possible.

I. Consider a design where i_r and i_s, $i_r < i_s$, are the indices of the positions occupied by the element α in the zeroth row and in the row $(i_u - i_r) \bmod m$, while $i_u \neq i_r$:

$$\begin{array}{c} \\ 0 \\ \\ (i_u - i_r) \bmod m \end{array} \begin{array}{c} \begin{array}{ccccc} & i_r & & i_s & \\ \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \alpha & \cdots \\ & \vdots & & & \\ \cdots & \alpha & \cdots & \alpha & \cdots \end{pmatrix} \end{array}.$$

Then in the zeroth row there is an element α in the positions i_r, i_s, i_u and $(i_u + i_s - i_r) \bmod m$. Note that

$$((i_u + i_s - i_r) \bmod m - i_u) \bmod m = i_s - i_r,$$

and hence the set D_α consists of less than $t_\alpha (t_\alpha - 1)$ elements.

Similarly, one may verify that if the matrix $\boldsymbol{M}$ is not bi-regular against the element β then the set D_β consists of less than $t_\beta (t_\beta - 1)$ elements.

II. Consider a design where i_r and i_s, $i_r < i_s$, are the indices of the positions occupied by the element α in the zeroth row and by the element β in the row $(j_u - i_r) \bmod m$,

while $j_u \neq i_r$:

$$\begin{array}{c} \\ 0 \\ (j_u - i_r)_{\bmod m} \end{array} \begin{array}{c} \begin{array}{cccc} & i_r & & i_s & \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \alpha & \cdots \\ & \vdots & & & \\ \cdots & \beta & \cdots & \beta & \cdots \end{pmatrix} \end{array}.$$

Then in the zeroth row there is the element β in the positions j_u and $(j_u + i_s - i_r)_{\bmod m}$. Note that

$$((j_u + i_s - i_r)_{\bmod m} - j_u)_{\bmod m} = i_s - i_r,$$

and hence the sets D_α and D_β have a common element $i_s - i_r$.

III. Consider a design where in the zeroth row and in the row $(i_s - i_r)_{\bmod m}$ in the columns i_r and j_u, $i_r < j_u$, there are the element α and the element β respectively, while $i_s \neq i_r$:

$$\begin{array}{c} \\ 0 \\ (i_s - i_r)_{\bmod m} \end{array} \begin{array}{c} \begin{array}{cccc} & i_r & & j_u & \end{array} \\ \begin{pmatrix} \cdots & \alpha & \cdots & \beta & \cdots \\ & \vdots & & & \\ \cdots & \alpha & \cdots & \beta & \cdots \end{pmatrix} \end{array}.$$

Then in the zeroth row there is the element α in the positions i_r and i_s and the element β in the positions j_u and $((i_s - i_r)_{\bmod m} + j_u)_{\bmod m}$. Note that

$$\left(((i_s - i_r)_{\bmod m} + j_u)_{\bmod m} - j_u\right)_{\bmod m} = (i_s - i_r)_{\bmod m},$$

and hence the sets D_α and D_β have a common element $(i_s - i_r)_{\bmod m}$.

The Theorem 1 is proved. ■

Corollary 2. Note the following particular case. An $m \times m$ circulant matrix with $t_\alpha > 1$ occurrences of an element α and $m - t_\alpha$ unique entries per row is bi-regular if and only if the set D_α comprises $t_\alpha (t_\alpha - 1)$ elements.

The next Lemma states that reducing the number of some element occurrences may result in a non-linear increase in the number of another element occurrences.

Lemma 3. Let $\boldsymbol{M}$ be a bi-regular circulant matrix over a subset $\mathscr{K}$ of some multiplicative group denoted by its zeroth row $(a_0, \ldots, a_{m-1})$. Suppose that there are $t_\alpha > 1$ and $t_\beta \geqslant 1$ positions occupied by an element α and an element β respectively. Then the decrease in the number t_α of element α occurrences by $k \in \{1, \ldots, t_\alpha - 1\}$ allows increase in the number t_β of element β occurrences by at most Δ_{t_β},

$$\Delta_{t_\beta} = \left\lfloor \frac{1}{2} + \left(\frac{1}{4} + t_\alpha (t_\alpha - 1) - (t_\alpha - k)(t_\alpha - k - 1) + t_\beta (t_\beta - 1) \right)^{1/2} \right\rfloor - t_\beta.$$

Proof. Given $t_\alpha > 1$, there exist $\binom{t_\alpha}{2}$ ways to select a pair of distinct indices of the positions occupied by the element α. A decrease in the number t_α by $k \in \{1, \ldots, t_\alpha - 1\}$ releases $\binom{t_\alpha}{2} - \binom{t_\alpha - k}{2}$ differences between two distinct indices that might be distributed to elements other than α. We now estimate Δ_{t_β} by which the number t_β of element β

occurrences might be increased while preserving the matrix $\boldsymbol{M}$ bi-regularity property. To achieve this objective, the following equation should be solved in integers for Δ_{t_β}:

$$\binom{t_\beta + \Delta_{t_\beta}}{2} - \binom{t_\beta}{2} = \binom{t_\alpha}{2} - \binom{t_\alpha - k}{2}.$$

Hence,

$$\Delta_{t_\beta} = \left\lfloor \frac{1}{2} + \left(\frac{1}{4} + t_\alpha \left(t_\alpha - 1\right) - \left(t_\alpha - k\right)\left(t_\alpha - k - 1\right) + t_\beta \left(t_\beta - 1\right) \right)^{1/2} \right\rfloor - t_\beta.$$

The Lemma 3 is proved. ■

Example 1. For $t = 4$ and $m = t(t-1) + 1 = 13$ consider a vector

$$(\alpha, \alpha, \beta, \gamma, \alpha, \delta, \alpha, \epsilon, \zeta, \eta, \theta, \iota, \kappa)$$

over a subset $\mathscr{K}$ of some multiplicative group. Note that distinct characters denote different group elements, and there are 10 distinct entries. One may verify that according to Theorem 1, this vector represents a bi-regular circulant matrix. If one element α is replaced by β, then there is a space for one more occurrence of β due to the fact that $\Delta_{t_\beta} = 2$. As an instance, we can take a vector

$$(\alpha, \alpha, \beta, \gamma, \alpha, \delta, \epsilon, \beta, \zeta, \beta, \eta, \theta, \iota)\,.$$

It can be verified that the new vector also represents a bi-regular circulant matrix.

3. Bi-regular circulant matrix layouts

Theorem 1 provides an efficient method of validation whether a circulant matrix is a bi-regular one. This method may be reduced to Algorithm 1.

Algorithm 1. Matrix bi-regularity validation algorithm

Require: circulant matrix $\boldsymbol{M} = \boldsymbol{M}_{m\times m}$.
Ensure: matrix $\boldsymbol{M}$ bi-regularity validation result.
$D := \varnothing$.
Reconstruct the set $\mathscr{K}$ of the elements of $\boldsymbol{M}$.
For all $e \in \mathscr{K}$:
    Find the indices $i_0, \ldots, i_{t_e - 1}$ of the positions occupied by an element e in one row, and count the number t_e.
    If $t_e > 1$, **then** construct the set D_e:

$$D_e = \left\{ (i - i') \bmod m : i, i' \in \{i_0, \ldots, i_{t_e-1}\}, i \neq i' \right\}.$$

        If $|D_e| < t_e\,(t_e - 1)$, **then return** «$\boldsymbol{M}$ is not bi-regular»,
            else if $D \cap D_e \neq \varnothing$, **then return** «$\boldsymbol{M}$ is not bi-regular»,
                else $D := D \cup D_e$.
Return «$\boldsymbol{M}$ is bi-regular».

The computational complexity of the algorithm 1 depends on the number $|\mathscr{K}|$ of different matrix elements and the number t_e of every distinct element e occurrences.

In essence, to positively validate the circulant matrix bi-regularity, algorithm 1 observes all the ordered pairs (i, i') for each element e, and the overall number of those pairs equals

$$2 \sum_{e \in \mathcal{K}} \binom{t_e}{2} = \sum_{e \in \mathcal{K}} (t_e^2 - t_e).$$

Note that, in contrast, negative validation result is obtained immediately after processing the first inappropriate matrix element. Thus, obtaining the negative validation result does not require observation of every ordered pair (i, i') for each element e, in general case.

Example 2. Consider a circulant $m \times m$ matrix with t_α occurrences of some element α per row, where $t_\alpha(t_\alpha - 1) = m - 1$. Other elements in each row occur only once. Then to positively validate such a matrix's bi-regularity, algorithm 1 must observe $m - 1$ ordered pairs of the distinct indices of the positions occupied by α.

As far as a general algorithm of the matrix bi-regularity validation is concerned, it takes to process all

$$\binom{m}{2}^2 = \frac{m^4 - 2m^3 + m^2}{4}$$

2×2 submatrices to ensure that a circulant matrix is bi-regular.

Now, an efficient method of the bi-regularity validation makes feasible the exhaustive search of arrays of variables that define bi-regular matrix layouts. Further, those layouts may be initialized by non-zero finite field elements. Following Table 1 gives a list of all non-equivalent arrays of the length $m = t_a\,(t_a - 1) + 1$ with $t_a \in \{2, 3, 4, 5, 6\}$ entries of some variable a which define bi-regular circulant matrix layouts. Here, two arrays are said to be non-equivalent if one is not a cyclic shifted representation of the other. These arrays are denoted by vectors $(i_0, \ldots, i_{t_a - 1})$ of indices of the variable a entries with $i_0 = 0$.

Table 1

t_a	m	Arrays of variables
2	3	$(0, 1)$
3	7	$(0, 1, 3)$ $(0, 2, 3)$
4	13	$(0, 1, 3, 9)$ $(0, 1, 4, 6)$ $(0, 1, 5, 11)$ $(0, 1, 8, 10)$
5	21	$(0, 1, 4, 14, 16)$ $(0, 1, 6, 8, 18)$
6	31	$(0, 1, 3, 8, 12, 18)$ $(0, 1, 3, 10, 14, 26)$ $(0, 1, 4, 6, 13, 21)$ $(0, 1, 4, 10, 12, 17)$ $(0, 1, 6, 18, 22, 29)$ $(0, 1, 8, 11, 13, 17)$ $(0, 1, 11, 19, 26, 28)$ $(0, 1, 14, 20, 24, 29)$ $(0, 1, 15, 19, 21, 24)$ $(0, 1, 15, 20, 22, 28)$

Remark 4. There are no arrays for $t_a = 7$ and $m = 43$ that denote bi-regular matrices.

Remark 5. Since for each array from Table 1 there are t_a occurrences of variable a and $m = t_a\,(t_a - 1) + 1$, all the variables different from a must occur only once conforming to Lemmas 1 and 2.

For software or hardware implementation arrays of the length $m \in \{8, 16, 32\}$ are preferable. Table 2 comprises a list of all non-equivalent arrays of the length $m \in \{8, 16\}$ with the maximal number t_a of the entries of some variable a for which the bi-regularity preserves. As in Table 1, these arrays are denoted by vectors $(i_0, \ldots, i_{t_a-1})$ of indices of the variable a entries with $i_0 = 0$.

Table 2

t_a	m	Arrays of variables
3	8	$(0, 1, 3)$ $(0, 1, 6)$
4	16	$(0, 1, 3, 7)$ $(0, 1, 3, 12)$ $(0, 1, 4, 6)$ $(0, 1, 4, 11)$ $(0, 1, 5, 7)$ $(0, 1, 5, 14)$ $(0, 1, 6, 13)$ $(0, 1, 10, 14)$ $(0, 1, 11, 13)$ $(0, 2, 5, 12)$ $(0, 2, 6, 13)$

Remark 6. There are no arrays for $t_a = 6$ and $m = 32$ producing bi-regular matrices.

4. Conclusion

The conducted survey of the circulant matrices comprises the following results. The upper bound of the number of some element occurrences for which the bi-regularity of a circulant matrix preserves was derived. Furthermore, necessary and sufficient conditions for the circulant matrix bi-regularity were proved, which made it possible to develop the efficient procedure of bi-regularity verification. We then managed to construct several bi-regular circulant matrix layouts of order up to 31 with the maximal number of some element occurrences. Besides, it was revealed that there are no layouts of order 32 with more than 5 occurrences of any element which yield a bi-regular matrix (and hence an MDS matrix).

Acknowledgement

The authors are grateful to the anonymous reviewer for the interest in the paper and the help in its improvement.

REFERENCES

1. *MacWilliams F. J. and Sloane N. J.* The Theory of Error-Correcting Codes, vol. 16. Elsevier, 1977.
2. *Segre B.* Curve razionali normali ek-archi negli spazi finiti. Ann. Matem. Pura Appl., 1955, vol. 39, no. 1, pp. 357–379.
3. *Ball S.* On sets of vectors of a finite vector space in which every subset of basis size is a basis. J. Europ. Math. Soc., 2012, vol. 14, no. 3, pp. 733–748.
4. *Junod P. and Vaudenay S.* Perfect diffusion primitives for block ciphers. LNCS, 2004, vol. 3357, pp. 84–99.
5. *Rozhkov M. I. and Malakhov S. S.* Experimental methods for constructing MDS matrices of a special form. J. Appl. Industr. Math., 2019, vol. 13, no. 2, pp. 302–309.
6. *Gupta K. C. and Ray I. G.* On constructions of circulant MDS matrices for lightweight cryptography. LNCS, 2014, vol. 8434, pp. 564–576.

7. *Li Y. and Wang M.* On the construction of lightweight circulant involutory MDS matrices. Intern. Conf. FSE, LNCS, 2016, vol. 9783, pp. 121–139.
8. *Cauchois V. and Loidreau P.* On circulant involutory MDS matrices. Designs, Codes and Cryptography, 2019, vol. 87, pp. 249–260.
9. *Kesarwani A., Sarkar S., and Venkateswarlu A.* Exhaustive search for various types of MDS matrices. IACR Trans. Symmetric Cryptology, 2019, pp. 231–256.
10. *Malakhov S. S. and Rozhkov M. I.* On construction of bi-regular circulant matrices, relating to MDS matrices. 2021 Intern. Conf. Engineering Technologies and Computer Science (EnT), IEEE, 2021, pp. 56–58.
11. *Zarankiewicz K.* Problem P 101. Colloq. Math., 1951, vol. 2, p. 131.
12. *Reiman I.* Über ein problem von K. Zarankiewicz. Acta Mathematica Academiae Scientiarum Hungarica, 1958, vol. 9, no. 3–4, pp. 269–273.

УДК 512.54 DOI 10.17223/20710410/56/3

SUPERPOSITIONS OF FREE FOX DERIVATIONS[1]

V. Roman'kov

Dostoevsky Omsk State University, Omsk, Russia
Siberian Federal University, Krasnoyarsk, Russia

E-mail: romankov48@mail.ru

Fox derivations are an effective tool for studying free groups and their group rings. Let F_r be a free group of finite rank r with basis $\{f_1, \ldots, f_r\}$. For every i, the partial Fox derivations $\partial/\partial f_i$ and $\partial/\partial f_i^{-1}$ are defined on the group ring $\mathbb{Z}[F_r]$. For $k \geqslant 2$, their superpositions $D_{f_i^\epsilon} = \partial/\partial f_i^{\epsilon_k} \circ \ldots \circ \partial/\partial f_i^{\epsilon_1}, \epsilon = (\epsilon_1, \ldots, \epsilon_k) \in \{\pm 1\}^k$, are not Fox derivations. In this paper, we study the properties of superpositions $D_{f_i^\epsilon}$. It is shown that the restrictions of such superpositions to the commutant F_r' are Fox derivations. As an application of the obtained results, it is established that for any rational subset R of F_r' and any i there are parameters k and ϵ such that R is annihilated by $D_{f_i^\epsilon}$.

Keywords: *free group, group ring, Fox derivations, annihilators, rational subsets.*

СУПЕРПОЗИЦИИ СВОБОДНЫХ ПРОИЗВОДНЫХ ФОКСА

В. А. Романьков

Омский государственный университет им. Ф. М. Достоевского, г. Омск, Россия
Сибирский федеральный университет, г. Красноярск, Россия

Дифференцирования Фокса являются эффективным инструментом исследования свободных групп и их групповых колец. Пусть F_r — свободная группа конечного ранга r с базисом $\{f_1, \ldots, f_r\}$. Для любого i частные дифференцирования Фокса $\partial/\partial f_i$ и $\partial/\partial f_i^{-1}$ определены на групповом кольце $\mathbb{Z}[F_r]$. Для $k \geqslant 2$ их суперпозиции $D_{f_i^\epsilon} = \partial/\partial f_i^{\epsilon_k} \circ \ldots \circ \partial/\partial f_i^{\epsilon_1}, \epsilon = (\epsilon_1, \ldots, \epsilon_k) \in \{\pm 1\}^k$ не являются дифференцированиями Фокса. В работе изучаются свойства суперпозиций $D_{f_i^\epsilon}$. Показано, что ограничения таких суперпозиций на коммутант F_r' являются дифференцированиями Фокса. В качестве приложения полученных результатов установлено, что для любого рационального подмножества R коммутанта F_r' и любого i существуют параметры k и ϵ, такие, что R аннулируется суперпозицией $D_{f_i^\epsilon}$.

Ключевые слова: *свободная группа, групповое кольцо, дифференцирования Фокса, аннуляторы, рациональные подмножества.*

1. Introduction

Let F_r be a free group of finite rank r with basis $\{f_1, \ldots, f_r\}$ and let $\mathbb{Z}[F_r]$ be the integral group ring. In the paper, we consider the partial free derivations introduced by Fox [1]. In our notation, these are defined as follows.

[1]The research was supported by the grant from the Russian Science Foundation (project no. 19-71-10017).

For $j = 1, \ldots, r$, the (left) Fox derivation associated with f_j is the linear map $D_j : \mathbb{Z}[F_r] \to \mathbb{Z}[F_r]$ satisfying the conditions

$$D_j(f_j) = 1,\ D_j(f_i) = 0 \ \text{ for } \ i \neq j,$$
$$D_j(uv) = D_j(u) + uD_j(v) \ \text{ for all } \ u, v \in F_r.$$

Obviously, an element $u \in F_r$ is trivial if and only if $D_i(u) = 0$ for all $i = 1, \ldots, r$. Also, note that for an arbitrary element g from F_r and for each $j = 1, \ldots, n$ we have $D_j(g^{-1}) = = -g^{-1}D_j(g)$. An excellent introduction to Fox's theory of derivations and their possible uses can be found in [2], see also [3, 4]. Free derivations of group rings were introduced by R. Fox for their use in knot theory. Later, the tools of free derivations began to be widely used in algebra. See, for example, the paper [5], where free derivations are applied to solve algorithmic problems in solvable groups. Nowadays, there are applications of Fox derivations to cryptography; namely, to generation of pseudorandom sequences over solvable groups. See, e.g., [6, Part II], where the ergodic theory for polynomials over solvable groups with operators is developed, or [7], where main results of the theory are announced (or an earlier expository paper [8]). The techniques used in these works also utilizes Fox derivations.

Also, for each $i = 1, \ldots, r$ there is a unique derivation D_j^- with respect to the inverse f_i^{-1} for which $D_j^-(f_j^{-1}) = 1$ and $D_j^-(f_i) = 0$ for any $i \neq j$. Then $D_j^-(f_j) = -f_j$. The *trivialization* homomorphism $\tau : \mathbb{Z}[F_r] \to \mathbb{Z}$ is defined on the generators of F_r by $\tau(f_i) = 1$ for all $i = 1, \ldots, r$ and extended linearly to the group ring $\mathbb{Z}[F_r]$.

The Fox derivations appear in another setting as well. Let ΔF_r denote the fundamental ideal of the group ring $\mathbb{Z}[F_r]$. It is a free left $\mathbb{Z}[F_r]$-module with a free basis consisting of $\{f_1 - 1, \ldots, f_r - 1\}$. This leads us to the following formula, which is called the *main identity* for the Fox derivations:

$$\sum_{i=1}^{r} D_i(\alpha)(f_i - 1) = \alpha - \tau(\alpha), \tag{1}$$

where $\alpha \in \mathbb{Z}[F_r]$. Conversely, if for any element $f \in F_r$ and $\alpha_i \in \mathbb{Z}[F_r]$ we have equality

$$\sum_{i=1}^{r} \alpha_i(f_i - 1) = f - 1,$$

then $D_i(f) = \alpha_i$ for $i = 1, \ldots, r$.

Frequently, the free derivations D_j are denoted by $\partial/\partial f_j$ for $j = 1, \ldots, r$. If $w = = w(v_1, \ldots, v_m)$ is a group word in variables $v_1, \ldots, v_m$, we consider the values of the formal derivations $\partial w/\partial v_i$, for $i = 1, \ldots, m$.

Proposition 1 (chain rule). If w and $v_1, \ldots, v_m$ are words in F_r, with $w = w(v_1, \ldots, v_m)$ and $v_i = v_i(f_1, \ldots, f_r)$ for $i = 1, \ldots, m$, then

$$\frac{\partial}{\partial f_i}(w(v_1, \ldots, v_m)) = \sum_{k=1}^{m} \partial w/\partial v_k \cdot \partial v_k/\partial f_i \text{ for any } i = 1, \ldots, r.$$

More generally, we call a linear map $D : \mathbb{Z}F_r \to \mathbb{Z}F_r$ the *Fox derivation* if D satisfies the property

$$D(uv) = D(u) + uD(v)$$

for all $u, v \in F_r$. Every such derivation has the form

$$D = \alpha_1 D_1 + \ldots + \alpha_n D_r,$$

where $\alpha_i = D(f_i)$ for $i = 1, \dots, r$. By definition, $(\alpha D)(u) = D(u)\alpha$ for any $\alpha \in \mathbb{Z}[F_r]$, $u \in F_r$. Conversely, we can define a derivation $D = \sum_{i=1}^{r} \alpha_i D_i$ for arbitrary tuple of elements $\alpha_i \in \mathbb{Z}[F_r]$.

In the paper, we consider superpositions of partial derivations with respect to some variable f_i and its inverse f_i^{-1}. Let $k \in \mathbb{N}$. For any i and $\epsilon = (\epsilon_1, \dots, \epsilon_k) \in \{\pm 1\}^k$ the corresponding superposition is

$$D_{f_i^\epsilon} = \partial/\partial f_i^{\epsilon_k} \circ \dots \circ \partial/\partial f_i^{\epsilon_1}.$$

Recall that for $g \in F_r$ we have $D_{f_i^\epsilon}(g) = \partial/\partial f_i^{\epsilon_k}(\partial/\partial f_i^{\epsilon_{k-1}}(\dots(\partial/\partial f_i^{\epsilon_1}(g))))$.

We use the following notation: $\sigma_i(v)$ — the sum of all exponents in which the variable x_i occurs in the word v, F_r' — the derived subgroup (commutant) of the group F_r, $F_r'(i) \leqslant F_r$ — the subgroup of all words v for which $\sigma_i(v) = 0$ $(i = 1 \dots, r)$ (i-commutant). Hence, $F_r' = \bigcap_{i=1}^{r} F_r'(i)$.

2. Basic results

Let $D = D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, be superposition of derivations on F_r. By definition, $\mathrm{Ann}(D)$ is the set of all elements $g \in F_r$ such that $D(g) = 0$ (annihilator of D on F_r).

Lemma 1. For any $i = 1, \dots, r$ and any finite set V of elements of F_r, there exists a number $k \in \mathbb{N}$ and a superposition of derivations $D = D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, such that $V \subseteq \mathrm{Ann}(D)$.

Proof. We use induction on the maximum number $m_i(V)$ of occurrences of $f_i^{\pm 1}$ in elements from V. If $m_i(V) = 0$, then we take $D = D_i$. Let us apply to each of the words $v \in V$ the superposition of derivations $D_{f_i^\epsilon}$ for $\epsilon = (1, -1)$. It is obvious that the value of m_i for the set of all resulting elements V' of F_r will become strictly smaller. By the inductive hypothesis, there exists a superposition of derivations $D' = D_{f_i^\varepsilon}$, $\varepsilon \in \{\pm 1\}^l$, for which all these elements belong to $\mathrm{Ann}(V')$. Therefore, we can take $\epsilon = (1, -1, \varepsilon) \in \{\pm 1\}^{l+2}$ and get $V \subseteq \mathrm{Ann}(D(V))$. ■

Proposition 2. Let $v \in F_r$, $\sigma_i(v) = 0$. Then for any superposition of derivations $D = D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, $k \in \mathbb{N}$, we have $\tau(D(v)) = 0$, i.e., the trivialization of $D(v)$ is zero.

Proof. We use induction on the (even) number $m_i(v)$ of occurrences of $f_i^{\pm 1}$ in v. If $m_i(v) = 0$, the statement is obvious. Let's represent v in the (reduced) form $v = u f_i^\nu z f_i^{-\nu} w$, $\nu \in \{\pm 1\}$, where z does not depend on $f_i^{\pm 1}$. Then we consider the corresponding reduced word $v' = u f_t w$, where $f_t \neq f_i$. By the induction hypothesis, the assertion of the proposition holds for v'.

Without changing the generality of reasoning, we assume that $\nu = 1$. We will sequentially perform derivations from the superposition D, starting from the element v. Consider the terms corresponding to the elements f_i and f_i^{-1} from the selected block $x = f_i z f_i^{-1}$. We have $u(1 - f_i z f_i^{-1})$ or $u(-f_i + f_i z)$. Further derivations do not change the factor u, results of derivations of the expression in brackets always has zero trivialization. Therefore, when calculating the trivialization, we can ignore the terms corresponding to the variables of the selected block.

There is a one-to one correspondence between the values of derivations corresponding to the occurences of $f_i^{\pm 1}$ for u and w in v and the corresponding occurences in v'. By the induction hypothesis, they all have zero trivialization. ■

Corollary 1. Any superposition of derivations of the form $D = D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, $k \in \mathbb{N}$, is derivation on $F_r'(i)$. Therefore, it is derivation on F_r'.

Proof. We use induction on k. For $k = 1$ the statement is true. Let it be true for superposition $D' = D_{f_i^{\epsilon'}}$, $\epsilon' \in \{\pm 1\}^{k-1}$. By induction we have

$$D'(uw) = D'(u) + uD'(w).$$

By Proposition 2, $\tau(D'(w)) = 0$. Therefore,

$$D(uv) = \partial/\partial f_i^{\epsilon_k}(D'(u)) + \partial/\partial f_i^{\epsilon_k}(u)(\tau(D'(u)) + u\partial/\partial f_i^{\epsilon_k}(D'(w)) = D(u) + uD(w).$$

Corollary 1 is proved. ■

Lemma 2. Let $v \in F_r$ and let $v = uw$ be the reduced presentation. For any superposition D of the form $D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, such that $D(v) = 0$ we have $D(w) = 0$, and if $\sigma_i(w) = 0$, then $D(u) = 0$.

Proof. The equality $D(w) = 0$ is obvious, since the results of computing D with respect to occurrences of elements $f_i^{\pm 1}$ in w can cancel only among themselves. The second assertion follows from Proposition 2. ■

3. Applications. Rational sets

Following R. H. Gilman [9], we define for a given group G the set $Rat(G)$ of all *rational subsets* of G as the closure of the set of all finite subsets of G under the rational operations: union, product, and generation of a submonoid (Kleene's star operation). It is known [9] that a subset R of a group G is rational in G if and only if R is accepted by a finite automaton over G. For basic information about rational sets in groups see [9, 10]. Special aspects of the theory are contained in [11 – 13].

Theorem 1. Let F_r be a free group and let R be a rational subset of F_r'. Then for any $i = 1, \ldots, r$ there is a superposition $D = D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, $k \in \mathbb{N}$, such that R belongs to $\mathrm{Ann}(D)$.

Proof. It is well known [9, 10] that any rational subset of an arbitrary group lies in a finitely generated submonoid. Let R lies in submonoid M generated by finite set $V \subseteq F_r'$. Let $V = \{v_1, \ldots, v_s\}$. By Lemma 1, there exists a superposition of derivations of the form $D = D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, such that $V \subseteq \mathrm{Ann}(D)$.

Let α be any element of M (in particular, $\alpha \in R$). We write α as the word $\alpha(v_1, \ldots, v_s)$. Let D_{v_i} be formal partial derivation. Then by (1) we have

$$\sum_{i=1}^{r} D_{v_i}(\alpha)(v_i - 1) = \alpha - \tau(\alpha).$$

Then by Proposition 2 we have

$$D\left(\sum_{i=1}^{r} D_{v_i}(\alpha)(v_i - 1)\right) = \sum_{i=1}^{r} D_{v_i}(\alpha)D(v_i) = 0.$$

Therefore, $D(\alpha - \tau(\alpha)) = D(\alpha) = 0$. ■

4. Conclusion

We have proved some results about superpositions of Fox partial derivations. In particular, we have established that the F_r'-restrictions of superpositions of the form $D_{f_i^\epsilon}$, $\epsilon \in \{\pm 1\}^k$, are Fox derivations. This allows us to use such superpositions to study the structure of subsets of the commutant F_r'. As an application, we have shown that any rational subset R of F_r' lies in an annihilator $\mathrm{Ann}(D_{f_i^\epsilon})$ for some $k \in \mathbb{N}$ and $\epsilon \in \{\pm 1\}^k$.

REFERENCES

1. *Fox R. H.* Free differential calculus I — Derivation in the free group ring. Ann. Math., 1953, vol. 57, pp. 547–560.
2. *Crowell R. H. and Fox R. H.* Introduction to Knot Theory. N.Y., Springer Verlag, 1963, X + 182 p.
3. *Timoshenko E. I.* Endomorfizmy i universal'nye teorii razreshimykh grupp [Endomorphisms and Universal Theories of Solvable Groups]. Novosibirsk, Novosibirsk State Technical University, 2011. 327 p. (in Russian)
4. *Roman'kov V. A.* Essays in Algebra and Cryptology. Solvable Groups. Omsk, Dostoevsky Omsk State University, 2017. 267 p.
5. *Myasnikov A., Roman'kov V., Ushakov A., and Vershik A.* The word and geodesic problems for free solvable groups. Trans. Amer. Math. Soc., 2010, vol. 362, no. 9, pp. 4655–4682.
6. *Anashin V. and Khrennikov A.* Applied Algebraic Dynamics (de Gruyter Expositions in Math., vol. 49). Berlin, N.Y., Walter de Gruyter GmbH & Co., 2009. 557 p.
7. *Anashin V.* Noncommutative algebraic dynamics: Ergodic theory for profinite groups. Proc. Steklov Institute of Mathematics, 2009, vol. 265, pp. 30–58.
8. *Anashin V. S.* Uniformly distributed sequences in computer algebra, or how to construct program generators of random numbers. J. Math. Sci., 1998, vol. 89, no. 4, pp. 1355–1390.
9. *Gilman R. H.* Formal languages and infinite groups. Geometric and Computational Perspectives of Infinite Groups, Minneapolis, MN, and New Brunswick, NJ, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 1994, vol. 25, pp. 27–51.
10. *Roman'kov V. A.* Ratsional'nye podmnozhestva v gruppakh [Rational subsets in groups]. Omsk, Dostoevsky Omsk State University, 2014. 176 p. (in Russian)
11. *Roman'kov V. A.* Polycyclic, metabelian or soluble of type $(FP)_\infty$ groups with Boolean algebra of rational sets and biautomatic soluble groups are virtually abelian. Glasgow Math. J., 2018, vol. 60, no. 1, pp. 209–218.
12. *Roman'kov V. A.* Rationality of verbal subsets in solvable groups. Algebra and Logic, 2018. vol. 57, no. 1, pp. 39–48.
13. *Roman'kov V. and Myasnikov A.* On rationality of verbal subsets in a group. Theory of Computing Systems, 2013, vol. 52, no. 4, pp. 587–598.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 004.056 DOI 10.17223/20710410/56/4

МЕТОДИКА ОЦЕНКИ БЕЗОПАСНОСТИ КРИПТОГРАФИЧЕСКИХ ПРОТОКОЛОВ[1]

А. Ю. Нестеренко, А. М. Семенов

Национальный исследовательский университет «Высшая школа экономики», Московский институт электроники и математики им. А. Н. Тихонова (МИЭМ НИУ ВШЭ), г. Москва, Россия

E-mail: anesterenko@hse.ru, amsemenov@hse.ru

Предлагается метод оценки безопасности криптографических протоколов, используемых для защиты информации как в информационно-телекоммуникационных сетях, так и в сетях «Интернета вещей». Описывается порядок оценки безопасности информационной системы, включающий в себя построение перечня угроз, модели угроз и детализации модели и возможностей нарушителя. Рассматривается понятие «свойство безопасности», приводится расширенный перечень указанных свойств, их классификация и формальная математическая модель. В рамках модели, для заданных свойств безопасности, предлагается метод получения численных значений показателей эффективности, зависящих от вероятности успеха и алгоритмической сложности решения ряда известных математических задач. Приводятся результаты применения предложенного метода к анализу стандартизируемых в Российской Федерации протоколов ESP и IKEv2 семейства IPSec.

Ключевые слова: *свойство безопасности, криптографический протокол, показатель эффективности защиты информации.*

METHODOLOGY FOR ASSESSING THE SECURITY OF CRYPTOGRAPHIC PROTOCOLS

A. Yu. Nesterenko, A. M. Semenov

National Research University "Higher school of economics", Tikhonov Moscow Institute of Electronics and Mathematics (MIEM NRU HSE), Moscow, Russia

This paper proposes a method for evaluating the security of cryptographic protocols used to protect information in telecommunication networks, as well as in networks of the "Internet of Things". The procedure for evaluation of information system security is described, including the construction of the list of threats, the threat model, and detailing of the model and the abilities of the intruder. The concept of security property is considered, the extended list of the specified properties, their classification and formal mathematical model are given. As part of the model, for given properties

[1]Исследование выполнено при финансовой поддержке РФФИ, проект № 19-37-90155.

of security, we propose a method for obtaining numerical values of performance parameters depending on the probability of success and algorithmic complexity of the solution of a number of known mathematical problems. In conclusion, the results of the application of the proposed method to the analysis of ESP and IKEv2 protocols of IPSec family standardized in the Russian Federation are presented.

Keywords: *security property, cryptographic protocol, information security performance indicator.*

Введение

В настоящее время не вызывает сомнений, что обеспечение безопасности информации, передаваемой в сетях связи (информационно-телекоммуникационных сетях, сетях «Интернета вещей» и т. п.) должно реализовываться при помощи криптографических механизмов — схем и протоколов. При этом высокий уровень защиты информации, обеспечиваемый криптографическими механизмами, является необходимым фактором для обоснования безопасности информационных (автоматизированных) систем, в состав которых входят сети связи.

Принятый в Российской Федерации порядок оценки безопасности информационных (автоматизированных) систем заключается в следующем:

1. Формируется модель угроз, создающих опасность нарушения безопасности передаваемой информации. Разработка модели угроз проводится для конкретной системы на основе базовых моделей, регламентируемых ФСТЭК и ФСБ России, а также государственными стандартами в области защиты информации.
   В настоящей работе будем основываться на базовой модели угроз, регламентируемой стандартом [1]. Данная модель включает в себя:
   - угрозу несанкционированного доступа к передаваемой информации (нарушение конфиденциальности);
   - угрозу несанкционированной передачи информации;
   - угрозу несанкционированного изменения информации (нарушение целостности);
   - угрозу отказа от факта отправки или приёма сообщения;
   - угрозу внесения вредоносного программного обеспечения;
   - угрозу отказа в обслуживании или предоставлении услуг (нарушение доступности).
2. Формируется модель нарушителя, содержащая совокупность возможностей, которые могут быть использованы при создании способов, подготовке и проведении атак, целью которых является реализация перечисленных угроз безопасности.
   Будем основываться на модели нарушителя, регламентируемой рекомендациями по стандартизации [2]. В данной модели инструментом реализации угроз безопасности являются проводимые нарушителем атаки на информационную систему и, в частности, на криптографические протоколы, обеспечивающие безопасность передачи информации. Каждая атака нарушителя может быть задана следующими характеристиками:
   а) объектом проведения атаки, безопасность которого должна обеспечиваться в течение определённого периода времени и/или определённого этапа жизненного цикла средства защиты информации;

б) возможностями, которые могут быть использованы при создании способов, подготовке и проведении атак; каждая возможность определяется сведениями, а также техническими средствами;
в) местом проведения атаки.

Применительно к анализу криптографических схем и протоколов понятие объекта атаки позволяет уточнить сформулированный выше перечень угроз. Это вызвано тем, что в качестве объектов атаки могут выступать параметры схем и протоколов, используемые для обеспечения криптографической защиты информации.

Возможности нарушителя в части воздействия на канал связи, по которому происходит обмен защищаемой информацией, принято описывать расширенной моделью Долева — Яо [3]. В рамках данной модели нарушитель обладает следующими возможностями:

— нарушителю известны форматы всех передаваемых сообщений;
— нарушитель может перехватить и получить содержимое любого сообщения от любого пользователя в сети связи;
— нарушитель может инициировать установление соединения с любым другим пользователем;
— нарушитель может изменять содержимое передаваемых пользователями сообщений и, в частности, посылать сообщения от имени другого пользователя;
— нарушитель может использовать все доступные ему комбинации сообщений или частей сообщений для формирования новых сообщений, в том числе расшифровывать и зашифровывать сообщения с помощью известных ему ключей шифрования, применяя любые доступные алгоритмы;
— нарушитель является полноценным пользователем сети, обладающим корректным собственным идентификатором и допустимым множеством ключевой информации;
— нарушитель может накапливать всю переданную в сеть связи информацию, проводить её анализ с применением специализированных технических средств и использовать результаты анализа для компрометации криптографических схем и протоколов;
— нарушитель может организовывать одновременное выполнение некоторого числа сессий одного и того же протокола защиты информации; сессии могут выполняться одновременно для различных участников протокола, при этом нарушитель может использовать информацию, передаваемую в ходе всех выполняемых сессий протокола.

Перечисленные методы реализации угроз безопасности принято называть «активными» атаками.

В модели Долева — Яо нарушитель может проводить также «пассивные» атаки на протокол, основанные на перлюстрации и последующем криптографическом анализе передаваемых в ходе выполнения протокола сообщений. При проведении пассивных атак предписанное заранее (регламентированное) выполнение прокола не меняется — нарушитель не изменяет передаваемые сообщения, не инициирует соединений и не вмешивается в логику взаимодействия пользователей сети.

Для усложнения рассматриваемой модели будем допускать, что нарушитель может компрометировать набор долговременных ключей любого потенциаль-

ного участника протокола, который не является участником атакуемой сессии выполнения протокола. Вариант нарушителя данного типа описан в [3, 4] и использован при построении атак на протокол Нидхема — Шредера [5].

3. Проводится исследование криптографических схем и протоколов, обеспечивающих безопасность сети связи, а также входящих в состав протокола криптографических преобразований. Целью исследований является определение численных значений одного или нескольких показателей эффективности [6], которые позволяют оценить уровень защищённости информационной системы. Система считается защищённой (безопасной), если значение показателя эффективности превышает величину, установленную нормативными, правовыми документами или требованиями по безопасности.

В результате оценки безопасности информационной (автоматизированной) системы мы должны получить ответы на следующие вопросы: что защищаем, от кого защищаем и как оценить уровень обеспечиваемой защиты?

Ответы на первые два вопроса могут быть получены с учётом области применения информационной (автоматизированной) системы и действующей в Российской Федерации нормативной базы. Однако единой методологии определения показателей эффективности и их значений применительно к криптографическим протоколам в настоящее время нет. В зарубежных публикациях принято использовать несколько подходов:

— базовую модель Белларе — Рогавея [7] и её модификации [8 – 11], в которых в качестве показателя эффективности может рассматриваться вероятность нарушения формального определения безопасного протокола; получение точных численных оценок показателя эффективности в данной модели не предполагается;
— модель Конетти — Кравчука [12] и её модификации [13 – 15], в которых в качестве показателя эффективности выступает величина отклонения от 1/2 вероятности различения двух моделей — практической модели протокола в рамках описанной выше модели нарушителя и «идеальной» модели протокола, реализующей обмен сообщениями по «идеальному» каналу связи без искажений и активного нарушителя.

Среди ранних работ по анализу криптографических протоколов стоит выделить [16, 17]. Более поздние обзоры зарубежных публикаций могут быть найдены в монографиях [18, 19].

В отечественных работах по анализу протоколов принято использовать два подхода:

— применение «практической стойкости», т. е. классического криптографического анализа для получения оценок стойкости используемых в протоколе криптографических примитивов [20, 21]; при данном подходе показателем эффективности служит минимальное из всех возможных значений трудоёмкости реализации известных атак на криптографические преобразования;
— применение теории «доказуемой стойкости», позволяющей исследовать безопасность протоколов в заданных вероятностных моделях поведения нарушителя с ограниченными вычислительными ресурсами; аналогично методу Канетти — Кравчука в качестве показателя эффективности в данном подходе выступает величина отклонения от 1/2 вероятности различения заданных параметров моделей от случайных равновероятно распределённых случайных величин [22, 23].

Настоящая работа использует первый подход к определению показателя эффективности защиты. Для расширения области его применения приводится способ построения формальной модели протокола — графа зависимостей между состояниями

субъектов взаимодействия. В п. 1 рассматривается понятие «свойства безопасности», позволяющего связать между собой угрозы безопасности и возможные атаки нарушителя (рис. 1). Приводится классификация свойств безопасности и их взаимосвязь между собой.

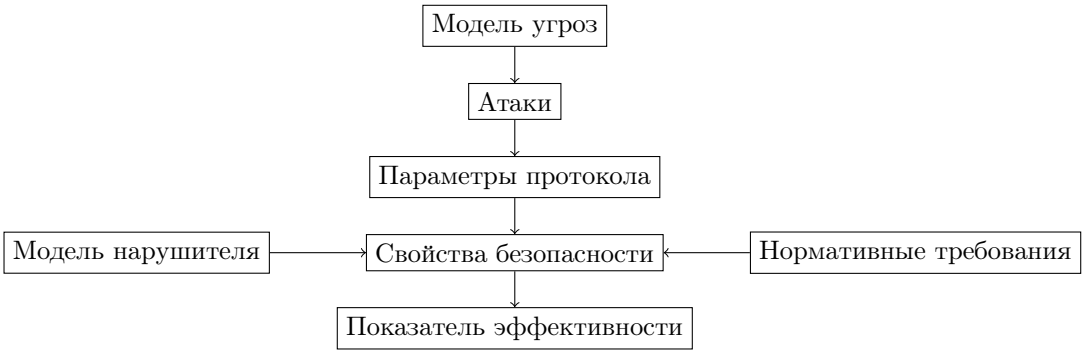

Рис. 1. Схема вычисления показателя эффективности

В п. 2 рассматривается формальная модель криптографического протокола и с её помощью моделируются базовые свойства безопасности. В п. 3 даётся формальное определение показателей эффективности защиты и предлагается способ определения их численных значений, основанный на трудоёмкости решения известных математических задач.

В п. 4 приводится методика получения численных значений показателя эффективности. Данная методика сложилась в ходе проведения исследований таких протоколов, как TLS 1.3 [24] и SP FIOT [25]; фрагменты методики рассматривались ранее в работах [26 – 28]. В п. 5 приводятся результаты применения предложенной методики к анализу стандартизированных в Российской Федерации протоколов ESP и IKEv2 семейства IPSec [29].

1. Свойства безопасности

Рассмотрим криптографические механизмы, получившие наибольшее распространение в современных информационных системах и обеспечивающие защищённый обмен информации с трёхсторонним участием — двух абонентов, обменивающихся информацией, и доверенного центра, обеспечивающего функции аутентификации участников взаимодействия. При этом участие доверенного центра в обмене информацией может быть косвенным, т. е. без отправки и получения сообщений.

Данные механизмы могут представлять собой совокупность, состоящую из нескольких схем и протоколов. В состав такой совокупности, как правило, входят:

— протокол односторонней, взаимной или многосторонней аутентификации участников информационного взаимодействия;
— протокол выработки общей для участников взаимодействия ключевой информации, действующей в рамках одной сессии информационного взаимодействия;
— транспортный протокол, предназначенный для передачи защищённой информации по каналам связи;
— процедуры выработки производной ключевой информации, контроля за временем и объёмом используемой ключевой информации;

— вспомогательные протоколы, предназначенные для передачи ошибок информационного взаимодействия, квитирования абонентов, инициализации процедуры выработки нового сессионного ключа и т. п.

В информационных системах «Интернета вещей» к этой совокупности могут добавляться протоколы взаимодействия различных сегментов общей сети связи. Проводимый анализ безопасности должен учитывать все элементы и выдавать единое значение показателя эффективности.

Инструментами, которые позволяют не только связать между собой угрозы безопасности и возможные атаки нарушителя, но и получать численные значения показателя эффективности, являются так называемые «свойства безопасности» (их называют также функциями-сервисами безопасности [30]).

Определение 1. Пусть π_0, $0 < \pi_0 \leqslant 1$, — заданное действительное число. Под свойством безопасности будем подразумевать свойство протокола прямо или косвенно обеспечивать невозможность реализации заданной угрозы с вероятностью, превышающей значение π_0.

Значение π_0 может быть равно 0,5, 0,1, 0,01, 0,001 и т. п. Оно определяет вероятность нарушения свойства безопасности и позволяет вывести из рассмотрения атаки с ничтожной вероятностью успеха, например случайное угадывание зашифрованного текста. Точное значение π_0 может определяться действующими требованиями по безопасности, моделью угроз безопасности передаваемой информации или рассчитываться с использованием риск-ориентированного подхода.

Свойства безопасности впервые вводятся в [7], а позднее расширяются в RFC 3552 [31] и в рамках проекта AVISPA [32]. Они рассматриваются также в ГОСТ Р ИСО/МЭК 27033-1:2011, разд. 7.3, и в работах [30, 28]. Расширим перечень из [32] и будем использовать следующие свойства безопасности:

С 1. *Свойство аутентификации субъекта* (*участника протокола*) *другим субъектом* (*участником протокола*) заключается в подтверждении одним субъектом подлинности другого субъекта, а также в получении гарантии того, что субъект, подлинность которого подтверждается, действительно принимает участие в выполнении текущей сессии протокола.
Свойство аутентификации субъекта может быть как односторонним, так и взаимным. В последнем случае свойство должно выполняться для всех участвующих во взаимодействии субъектов. Данное свойство содержится в [30, разд. 3; 31, п. 2.1.3; 32, свойство G1].

С 2. *Свойство аутентификации сообщения* заключается в подтверждении подлинности источника сообщения и целостности передаваемого сообщения.
Подлинность источника сообщения означает, что протокол должен обеспечивать гарантии того, что полученное сообщение или его часть были созданы участником взаимодействия в ходе выполнения текущей сессии протокола в некоторый момент времени, предшествующий получению сообщения. Фактически в рамках данного свойства сообщение однозначно связывается со своим источником (субъектом, отправившим сообщение), а выполнение свойства гарантирует, что сообщение не было искажено, в частности подделано нарушителем, при передаче по каналам связи. Данное свойство содержится в [30, разд. 3; 31, п. 2.1.2; 32, свойство G2].

С 3. *Свойство целостности сообщений* заключается в том, что получатель сообщения обладает возможностью проверить, что полученные им данные (или их

часть) не были модифицированы, уничтожены и являются теми же самыми данными, что послал отправитель. Данное свойство содержится в [30, разд. 2; 31, п. 2.1.2].

С 4. *Свойство защиты от повторов* заключается в том, что один раз корректно принятое участником протокола сообщение не должны быть принято повторно. В зависимости от протокола данное свойство может быть сформулировано в виде одного из следующих требований:
   — должна быть обеспечена гарантия того, что сообщение выработано в рамках текущей сессии протокола;
   — должна быть обеспечена гарантия того, что сообщение выработано в рамках заданного интервала времени;
   — сообщение не было принято ранее.

   В отечественной литературе данное свойство часто называют *свойством невозможности навязывания ложных сообщений*, подразумевая под этим защиту как от повторного принятия истинных сообщений, так и от подделанных нарушителем сообщений (свойство С 2). Данное свойство содержится в [30, разд. 3; 32, свойство G3].

С 5. *Свойство неявной аутентификации получателя* заключается в том, что протокол должен обладать средствами, гарантирующими, что отправленное сообщение может быть прочитано только теми участниками, для которых оно предназначено. Только законные авторизованные участники должны иметь доступ к данной информации, многоадресным сообщениям или групповому взаимодействию. Данное свойство содержится в [32, свойство G4].

С 6. *Свойство групповой аутентификации* заключается в том, что законные авторизованные члены заранее определённой группы пользователей могут аутентифицировать источник и содержание информации или группового сообщения. Сюда также входят протоколы, в которых участники группового взаимодействия не доверяют друг другу. Данное свойство содержится в [30, разд. 3; 32, свойство G5].

С 7. *Свойство аутентификации субъекта* (*участника протокола*) *доверенной третьей стороной*. В протоколах, явно реализующих взаимодействие участников с доверенной третьей стороной, данное свойство эквивалентно первому из перечисленных свойств.

   В случае использования инфраструктуры открытых ключей данное свойство может выполняться косвенно, путём заверения открытых ключей участников взаимодействия электронной подписью удостоверяющего (доверенного) центра; при этом привязка аутентификации субъекта к какой-либо сессии протокола не может быть обеспечена. Данное свойство содержится в [32, свойство G6].

С 8. *Свойство конфиденциальности ключа* предполагает, что в ходе информационного взаимодействия значение ключа не может стать известным нарушителю, а также легитимным пользователям информационной системы, для которых данный ключ не предназначен. Данное свойство может применяться как к исходной ключевой информации, так и к производным сессионным ключам.

С 9. *Свойство аутентификации ключа* предполагает, что один из участников взаимодействия получает подтверждение того, что никакой другой участник, кроме заранее определённого второго участника и, возможно, доверенного центра, не может обладать секретным ключом, выработанным в ходе выполнения протокола. Данное свойство содержится в [30, разд. 3; 32, свойство G7].

С 10. *Свойство подтверждения ключа* заключается в том, что один из участников взаимодействия получает подтверждение того, что второй участник (или группа участников) действительно обладает заданным секретным ключом и/или имеет доступ к информации, необходимой для выработки заданного секретного ключа. Данное свойство содержится в [30, разд. 3; 32, свойство G8].

С 11. *Свойство стойкости при компрометации производных ключей* состоит в том, что компрометация производных ключей, т. е. ключей, используемых непосредственно для шифрования и имитозащиты передаваемой информации, не приводит к нарушению других свойств безопасности как в рамках текущей, так и в других сессиях протокола, в частности к компрометации производных ключей, выработанных ранее или планируемых к выработке в дальнейшем.
В литературе данное свойство часто называют защитой от «чтения вперед/назад» или используют термин «perfect forward secrecy». Данное свойство содержится в [30, разд. 3; 32, свойство G9].

С 12. *Свойство стойкости при компрометации ключа аутентификации* состоит в том, что компрометация долговременного ключа аутентификации не приводит к нарушению конфиденциальности информации, переданной до момента компрометации ключа, а в случае пассивного нарушителя — и к нарушению конфиденциальности информации, передаваемой после завершения текущей сессии протокола. В литературе данное свойство иногда называют защитой от «чтения назад».

С 13. *Свойство формирования новых ключей* заключается в том, что протокол, обладающий данным свойством, позволяет формировать уникальные сессионные и/или производные ключи для каждой сессии протокола. Данное свойство содержится в [32, свойство G10].

С 14. *Свойство защиты от навязывания ключевых значений* гарантирует, что ни один из участников протокола не может навязать значение общего секретного, сессионного или производного ключа по своему выбору другому участнику протокола.

С 15. *Свойство защиты от навязывания параметров безопасности* гарантирует, что используемые в ходе выполнения протокола или согласуемые на этапе установления соединения параметры безопасности не могут быть навязаны нарушителем. В качестве параметров безопасности могут выступать наборы используемых криптографических преобразований, численные параметры алгоритмов и алгебраических структур, в которых выполняется протокол, случайные значения, вырабатываемые в ходе выполнения протокола и т. п. Данное свойство содержится в [32, свойство G11].

С 16. *Свойство конфиденциальности* заключается в том, что данные, передаваемые в ходе информационного взаимодействия, не могут стать известными нарушителю и/или легитимным участникам, для которых они не предназначены. Данное свойство содержится в [30, разд. 3; 31, п. 2.1.1; 32, свойство G12]. Легко видеть, что нарушение свойства конфиденциальности ключевой информации (С 8) приводит к нарушению конфиденциальности передаваемых данных.

С 17. *Свойство инвариантности отправителя* заключается в том, что на протяжении выполнения всего протокола получатель сообщений сохраняет уверенность в том, что источник сообщения остался тем же, что и источник, с которым было начато взаимодействие (сессия протокола). Данное свойство содержится в [32, свойство G16].

С 18. *Свойство анонимности субъекта* (*участника протокола*) состоит в том, что нарушитель, осуществляющий перехват сообщений, не должен иметь возможность связать сообщения одного из участников с самим участником или его идентификатором. Данное свойство содержится в [32, свойство G13].

С 19. *Свойство анонимности субъекта для других участников* заключается в том, что каждый участник взаимодействия не должен иметь возможность узнать реальную личность других участников, а должен взаимодействовать с их псевдонимом или случайным идентификатором. Данное свойство содержится в [32, свойство G14].

С 20. *Свойство защищённости от атак «отказ в обслуживании»* подразумевает, что реализующее протокол средство защиты информации обеспечивает алгоритмические, технические и организационно-штатные меры защиты от указанного типа атак. Данное свойство содержится в [32, свойство G15].

Теоретическое исследование протокола может лишь проверить наличие алгоритмических мер, обеспечивающих защиту от данного класса атак, а также наличие эксплуатационной документации, содержащей описание технических и организационно-штатных мер защиты. В рамках предлагаемой методики представляется возможным получить лишь тривиальное численное значение показателя эффективности для данного свойства.

С 21. *Свойство защищённости от утечек по скрытым* (*логическим*) *каналам* подразумевает, что протокол содержит реализацию алгоритмических мер защиты от атак, реализуемых нарушителем путём применения непредусмотренных коммуникационных каналов передачи информации. Отметим, что современные транспортные протоколы, такие, как ESP, IPSec или ADTP FIOT, содержат ряд мер, предназначенных для обеспечения данного свойства.

Классификация угроз безопасности, реализуемых с использованием скрытых каналов, модель нарушителя и перечень мер защиты информационной системы от атак с использованием скрытых каналов должны разрабатываться на основе стандартов [33, 34]. Получение численных оценок показателей эффективности мер защиты от скрытых логических каналов выходит за рамки настоящей работы. Отдельные результаты в данном направлении получены в работах [35 – 37].

С 22. *Свойство защищённости от KCI-атак.* Под KCI-атакой (атакой имперсонификации при компрометации долговременного секретного ключа) понимается атака, при выполнении которой нарушитель, получивший доступ к долговременному секретному ключу участника, может выдать себя перед ним за любого другого участника в рамках текущей или будущей сессии выполнения протокола. Свойство считается выполненным, если KCI-атака невыполнима. Данное свойство описано в [10].

С 23. *Свойство защищённости от UKS-атак.* Под UKS-атакой понимается последовательность действий нарушителя, в результате которой законные авторизованные участники в процессе информационного взаимодействия вырабатывают общий ключ, но один из участников считает, что он выработал общий ключ с третьим участником (навязанным нарушителем в ходе выполнения протокола). При этом компрометации общего ключа как таковой не происходит, но нарушается требование аутентификации участников. Свойство считается выполненным, если подобная ситуация невозможна. Данное свойство описано в [38, 39]

С 24. *Свойство невозможности отказа от совершённых действий* представляет собой возможность проследить за всеми действиями участника взаимодействия.

Согласно Р 1323565.1.012-2017, разд. 6.1.14, данное свойство должно обеспечиваться средством криптографической защиты информации, реализующим криптографический протокол. Данное свойство содержится в [32, свойство G17].

С 25. *Свойство доказательства происхождения* заключается в неоспоримом доказательстве отправки сообщения. Данное свойство содержится в [32, свойство G18].

С 26. *Свойство доказательства доставки* заключается в неоспоримом доказательстве получения сообщения. Данное свойство содержится в [32, свойство G19].

С 27. *Свойство целостности множества состояний* (*криптографическое связывание состояний*) заключается в том, что все участники информационного взаимодействия после выполнения протокола (или его части) в рамках одного сеанса связи имеют одинаковое представление обо всех участниках этого сеанса и выполняемых ими ролях, а также о состоянии выполнения протокола. Данное свойство описано в рекомендациях [40 – 42].

Можно провести классификацию свойств безопасности по объектам применения, влияющим на безопасность исследуемого криптографического протокола (табл. 1).

Таблица 1

Свойства безопасности по объектам применения

Объект применения	Свойства безопасности
Аутентификация	С 1, С 2, С 5, С 6, С 7, С 9
Целостность	С 3, С 27
Ключи	С 8, С 10, С 11, С 12, С 13, С 14
Субъект взаимодействия	С 17, С 18, С 19, С 24
Атаки нарушителя	С 4, С 15, С 20, С 21, С 22, С 23
Защищаемые данные	С 16, С 25, С 26

Отметим, что для большинства используемых на практике криптографических протоколов все перечисленные свойства безопасности не могут быть выполнены одновременно. Примерная классификация свойств безопасности, которые могут обеспечиваться протоколами с различными целевыми назначениями, приведена в табл. 2 (см. также [30]).

Таблица 2

Свойства безопасности по целевому назначению криптографических протоколов

Класс протоколов	Свойства безопасности
Протоколы обеспечения целостности сообщения	С 4, С 10, С 13, С 3, С 22, С 23
Протоколы на основе цифровой подписи	С 1, С 2, С 8, С 9, С 11, С 12, С 17, С 22, С 23
Протоколы на основе цифровой подписи вслепую	С 1, С 5, С 11, С 12, С 19, С 22, С 23
Протокол односторонней аутентификации	С 1, С 2, С 8, С 9, С 11, С 12, С 17, С 22, С 23
Протокол взаимной аутентификации	С 1, С 2, С 6, С 8, С 9, С 11, С 12, С 17, С 22, С 23
Протокол групповой аутентификации	С 1, С 2, С 6, С 9, С 11, С 12, С 17, С 22, С 23
Протоколы конфиденциальной передачи	С 13, С 15, С 16, С 3, С 22, С 23,
Протоколы распределения ключей	С 1, С 2, С 8, С 9, С 22, С 23
Протоколы выработки общего ключа	С 1, С 2, С 4, С 8, С 9, С 10, С 11, С 22, С 23

Следует отметить, что на практике сложно отнести криптографический протокол к тому или иному классу, поскольку в большинстве случаев протоколы обеспечивают выполнение свойств, характерных для нескольких целевых назначений.

Для построения формальной модели свойств безопасности полезно разбить сформулированные свойства на два больших класса:

— базовые свойства, выполнение которых зависит от сложности решения математических задач, используемых в криптографических преобразованиях;
— производные свойства, являющиеся комбинацией базовых и других производных свойств.

Зависимость между свойствами безопасности представлена в табл. 3.

Таблица 3

Зависимости между свойствами безопасности

Свойство	Зависимость
С 1 — аутентификации участника протокола другим участником	Базовое
С 2 — аутентификации сообщения	С 1, С 3
С 3 — целостности сообщений	Базовое
С 4 — защиты от повторов	Базовое
С 5 - неявной аутентификации получателя	С 1, С 9
С 6 — групповой аутентификации	С 1, С 9
С 7 — аутентификации субъекта доверенной третьей стороной	С 1
С 8 — конфиденциальности ключа	Базовое
С 9 — аутентификации ключа	С 1, С 2, С 10, С 15
С 10 — подтверждения ключа	Базовое
С 11 — стойкости при компрометации производных ключей	С 13
С 12 — стойкости при компрометации ключа аутентификации	С 13
С 13 — формирования новых ключей	С 15
С 14 — защиты от навязывания ключевых значений	С 1, С 3
С 15 — защиты от навязывания параметров безопасности	С 1, С 2, С 3
С 16 — конфиденциальности	С 3, С 9, С 10
С 17 — инвариантности отправителя	С 1, С 9
С 18 — анонимности субъекта	Базовое
С 19 — анонимности субъекта для других участников	Базовое
С 20 — защищённости от атак «отказ в обслуживании»	Базовое
С 21 — защищённости от утечек по скрытым (логическим) каналам	Базовое
С 22 — защищённости от KCI-атак	С 1, С 9, С 10, С 12, С 13, С 14
С 23 — защищённости от UKS-атак	С 1, С 9, С 10, С 15, С 27
С 24 — невозможности отказа от совершенных действий	С 25, С 26, С 27
С 25 — доказательства происхождения	С 1, С 2, С 9
С 26 — доказательства доставки	С 9, С 10
С 27 — целостности множества состояний	С 1, С 17, С 22, С 23

Указанные зависимости позволяют свести исследование большого числа свойств безопасности к малому числу базовых свойств.

Отметим также, что в криптографических механизмах, представляющих собой совокупность нескольких протоколов, свойства безопасности могут наследоваться. Например, транспортный протокол, реализующий только шифрование и имитозащиту передаваемой информации, сам по себе не обеспечивает свойство аутентификации субъектов взаимодействия, однако он может его наследовать в случае использования ключевой информации, предварительно полученной в ходе протокола выработки ключей с аутентификацией участников. Подобная ситуация характерна для многих современных криптографических механизмов, включая TLS, IPSec или SP FIOT.

2. Формализация модели безопасности и моделирование свойств безопасности

Данное выше описание свойств безопасности носит качественный характер и не позволяет предъявить какой-либо способ определения показателя эффективности за-

щиты. Формализуем модель криптографического протокола и с её помощью опишем свойства безопасности как составные части исследуемого протокола.

Предлагаемая модель отталкивается от атомарного подхода к описанию протоколов и является моделью дискретной динамической системы. Состояния системы определяются в некоторые моменты времени и, согласно спецификации протокола, зависят от функций перехода в следующее состояние и данных, поступающих из канала связи.

Определение 2. Обозначим символом $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ булево множество, элементы которого принимают значения «истина» или «ложь». Символом $\mathbb{V}_\infty^*$ обозначим множество двоичных последовательностей произвольной конечной длины, включая последовательность длины 0 (обозначим её $\varnothing$). Символом $\mathbb{V}_\infty$ обозначим множество последовательностей ненулевой длины.

Пусть $a = (\alpha, \sigma)$ — некоторая абстрактная ячейка памяти. Будем говорить, что ячейка характеризуется:

1) значением $\alpha \in \mathbb{V}_\infty^*$; неопределённому значению ячейки a соответствует символ $\varnothing$;
2) подтверждением $\sigma \in \mathbb{B}$: $\sigma = \texttt{true}$ соответствует подтверждённому значению, $\sigma = \texttt{false}$ — неподтверждённому значению ячейки a.

Понятие «подтверждения» ячейки вводится для того, чтобы формализовать уверенность субъекта, владеющего ячейкой $a = (\alpha, \sigma)$, в том, что значение α, содержащееся в подтверждённой ячейке, истинно, а не вычислено ошибочно или подделано и/или навязано нарушителем.

Определение 3. Пусть $\{t_k\}_0^\infty$ — монотонно возрастающая последовательность натуральных чисел, где $k = 0, 1, \ldots, k_{\max}$ для некоторого натурального значения $k_{\max}$, определяемого спецификацией протокола.

Для субъекта A будем называть его состоянием в момент времени t_k множество ячеек памяти

$$A(t_k) = \{a_1, \ldots, a_{n(A)} : \; a_i = \big(\alpha_i(t_k), \sigma_i(t_k)\big)\},$$

значения и подтверждения которых могут изменяться с изменением момента времени. Количество ячеек памяти $n(A)$ зависит от роли субъекта и определяется спецификацией протокола.

У различных субъектов точные значения временных меток t_k могут отличаться. Можно считать, что t_0 — это время начальной инициализации состояния субъекта, а $t_1, t_2, \ldots$ — времена отправки и получения сообщений из канала связи.

В ряде случае, например в транспортных протоколах, реализуется только процедура отправки (получения) сообщений, а значение величины $k_{\max}$ может быть, формально, не ограничено. Однако существующие в Российской Федерации требования по ограничению объёма зашифровываемой на одном ключе информации (см. Р 1323565.1.012-2017) накладывают ограничения на число передаваемых сообщений и, как следствие, на количество возможных состояний $k_{\max}$.

Определение 4. Будем говорить, что модель протокола определена, если для каждого субъекта:

— определено множество ячеек памяти, образующих изменяемое в ходе выполнения протокола состояние;
— определено количество возможных состояний,

а также в соответствии со спецификацией протокола определены функции перехода из одного состояния в другое, позволяющие однозначно определить значение и подтвер-

ждение каждой ячейки памяти, т. е. для всех $k = 0, 1, \ldots, k_{\max}$ и всех $i = 1, \ldots, n(A)$ определены:

1) целые неотрицательные числа l_k;
2) отображения

$$\texttt{var}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^*)^{n(A)+l_k} \to \mathbb{V}_\infty^*,$$
$$\texttt{conf}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^* \times \mathbb{B})^{n(A)} \times (\mathbb{V}_\infty^*)^{l_k} \to \mathbb{B},$$

такие, что

$$\alpha_i(t_{k+1}) = \texttt{var}_{i,k}(\alpha_1(t_k), \ldots, \alpha_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)),$$
$$\sigma_i(t_{k+1}) = \texttt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)),$$

где $i = 1, \ldots, n(A)$, а значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ рассматриваются как реализации l_k случайных величин, принимающих значения из $\mathbb{V}_\infty^*$ в момент времени t_k.

Полагаем, что в подавляющем большинстве случаев введённые отображения $\texttt{var}_{i,k}$ и $\texttt{conf}_{i,k}$ задаются тривиальными соотношениями

$$\begin{array}{lcl} l_k & = & 0, \\ \alpha_i(t_{k+1}) & = & \texttt{var}_{i,k}(\alpha_1(t_k), \ldots, \alpha_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \quad = \quad \alpha_i(t_k), \\ \sigma_i(t_{k+1}) & = & \texttt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \quad = \quad \sigma_i(t_k) \end{array}$$

при $k \in \{1, \ldots, k_{\max}\}$. В данном случае аргументы функций $\texttt{var}_{i,k}$ и $\texttt{cont}_{i,k}$, отличные от $\alpha_i(t_k)$ и $\sigma_i(t_k)$ соответственно, являются несущественными, т. е. не влияют на возвращаемое значение.

Для остальных случаев поясним смысл, который вкладывается в случайные значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$:

1) Случай $l_k = 0$ описывает автономное изменение состояния, которое субъект выполняет без какого-либо влияния извне. Такое изменение может использоваться для детализации спецификации протокола, например для изменения или подтверждения состояний элементов ключевой системы.
2) В ряде протоколов для аутентификации субъектов взаимодействия или выработки общей ключевой информации требуется генерация случайных значений; именно эти значения выступают в качестве величин $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, изменяющих состояние участника протокола (для подавляющего числа протоколов в этом случае можно считать, что $l_k = 1$).
3) Во всех протоколах субъект взаимодействия обрабатывает данные, поступающие из канала связи и рассматриваемые как случайные величины $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, принимающие значения в своей области определения.

Введённые отображения $\texttt{var}_{i,k}$ формально должны быть представлены в виде двухэтапной процедуры. На первом этапе должны быть определены отображения

$$\texttt{validate}_{j,k}(\xi_j(t_k)) : \mathbb{V}_\infty^* \to \mathbb{B}, \quad j = 1, \ldots, l(k),$$

определяющие принадлежность значения $\xi_j(t_k)$ заданной области определения. На втором этапе отображения $\texttt{var}_{i,k}$ должны реализовывать определяемые спецификацией протокола функции

$$\texttt{evaluate}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^* \times \mathbb{B})^{n(A)} \times (\mathbb{V}_\infty^*)^{l_k} \to \mathbb{V}_\infty^*,$$

изменяющие значение переменой a_i.

В случае, если одна из функций $\texttt{validate}_{j,k}$ возвращает `false`, то переход в следующее состояние должен блокироваться:

— если это указано в спецификации протокола, сообщение, содержащее хотя бы одно из таких значений $\xi_j(t_k)$, должно отбрасываться — для транспортных протоколов, либо протокол должен прекращать свое выполнение — для протоколов выработки ключа;
— если в спецификации протокола область допустимых значений для $\xi_j(t_k)$ не определена, то считаем, что это ошибка синтеза протокола, приводящая к нарушению его безопасности с вероятностью 1.

Если спецификацией протокола определена область допустимых значений для $\xi_j(t_k)$, то функции $\texttt{validate}_{j,k}$ не влияют на оценку безопасности. Далее мы выводим их из рассмотрения, полагая функции $\texttt{var}_{i,k}$ и $\texttt{evaluate}_{i,k}$ эквивалентными. Вместе с тем при практической реализации протокола функции $\texttt{validate}_{j,k}$ играют существенную роль, поскольку их отсутствие приводит к появлению уязвимостей в программном или аппаратном обеспечении средства защиты информации.

Функции $\texttt{conf}_{i,k}$ предназначены для подтверждения того, что значение $\alpha_i(t_k)$ является истинным, а величины, использованные для определения или формирования значения $\alpha_i(t_k)$, не были искажены или навязаны нарушителем в процессе обмена информацией по каналам связи.

Примером функции подтверждения могут служить функции проверки имитовставки или электронной подписи, которые позволяют гарантировать истинность подтверждаемых значений при помощи криптографических преобразований. При этом допускается, что одна функция $\texttt{conf}_{i,k}$ может подтверждать истинность нескольких ячеек $\alpha_{i_1}(t_k), \alpha_{i_2}(t_k), \ldots$, если все они одновременно являются аргументами функции $\texttt{conf}_{i,k}$, например, при проверке имитовставки проверяется истинность как сообщения, так и используемого секретного ключа. Далее всегда предполагается, что одним из аргументов функции $\texttt{conf}_{i,k}$ является некоторая ключевая информация, определяемая перед началом протокола (исходная ключевая информация) или вырабатываемая в ходе его выполнения.

Определение 5. Зафиксируем индекс $i \in \{1, \ldots, n(A)\}$ и момент времени t_k, $k \in \{1, \ldots, k_{\max}\}$.

1) Будем говорить, что значение ячейки $a_i = (\alpha_i(t_k), \sigma_i(t_k))$ подтверждено в момент времени t_k, если $\sigma_i(t_k) = \texttt{true}$.
2) Будем говорить, что значение ячейки a_i подтверждено косвенно, если $\sigma_i(t_k) = \texttt{false}$ и значение $\alpha_i(t_k)$ определено равенством

$$\alpha_i(t_k) = \texttt{var}_{i,k-1}(\alpha_{i_1}(t_{k-1}), \ldots, \alpha_{i_{s_i}}(t_{k-1})), \quad i_1, \ldots, i_{s_i} \in \{1, \ldots, n(A)\},$$

т. е. зависит только от существенных значений $\alpha_{i_1}(t_{k-1}), \ldots, \alpha_{i_{s_i}}(t_{k-1})$, таких, что

$$\sigma_{i_1}(t_{k-1}) = \cdots = \sigma_{i_{s_i}}(t_{k-1}) = \texttt{true}.$$

Можно предположить, что криптографический протокол является безопасным для субъекта A в момент времени t_k, где $k \in \{1, \ldots, k_{\max}\}$, если значения всех ячеек памяти состояния $A(t_k)$ являются либо подтверждёнными с использованием криптографических преобразований или выработаны самим субъектом, либо подтверждены косвенным образом. Однако, как мы покажем позднее, это предположение является необходимым, но не достаточным условием безопасности протокола.

В отличие от большинства других подходов к моделированию криптографических протоколов, предложенная модель ориентирована не на поиск возможных действий нарушителя и построение атак на протокол, а на поиск и построение графа зависимостей между всеми ячейками состояния субъекта, позволяющими проследить состояние ячеек памяти и подтвердить их «истинность», начиная с некоторого шага выполнения протокола (момента времени t_k).

Если спецификация протокола допускает существование неподтверждаемых переменных, то поиск возможных атак должен производиться с использованием автоматических верификаторов, таких, как Avispa [32], Scyther [43] или Proverif [44].

В рамках сформулированной модели вопрос о том, какие именно ячейки памяти должны входить в состояние субъекта, не конкретизируется. Уточнение перечня используемых ячеек памяти, а также построение зависимостей между ними должны производиться при анализе протокола (см. далее п. 4) и учитывать предъявляемые к протоколу свойства безопасности. Для упрощения данной процедуры проведём формализацию ряда базовых свойств безопасности (см. табл. 3).

2.1. Свойство аутентификации субъекта

Напомним, что с 2020 г. вопросы идентификации и аутентификации субъектов взаимодействия в Российской Федерации должны решаться с учётом ГОСТ Р 58833-2020 [45]. Согласно данному стандарту, при взаимодействии сторон с целью доступа к информации должны быть выполнены следующие процедуры:

1) первичная идентификация, в ходе которой регистрирующей стороной (доверенным центром) субъекту доступа должен присваиваться уникальный идентификатор $ID \in \mathbb{V}_\infty$;
2) вторичная идентификация, целью которой является опознавание субъекта доступа, т. е. предъявление субъектом присвоенного ранее идентификатора ID при попытке доступа к информации; выполнение вторичной идентификации производится субъектом, предоставляющим доступ к информации, — в нашем случае другим участником взаимодействия;
3) аутентификация субъекта доступа, в которую должны входить действия по проверке подлинности субъекта доступа, а также принадлежности субъекту доступа предъявленного идентификатора и аутентификационной информации.

Аутентификация субъекта доступа, согласно [45], может осуществляться с использованием нескольких факторов:

— фактора знания определённой информации, например секретного ключа или пароля;
— фактора владения определённым предметом;
— биометрическим фактором, описывающим определённые характеристики аутентифицируемого субъекта.

Поскольку при разработке криптографических протоколов принято использовать только фактор знания ключевой информации, мы должны дополнить положения стандарта [45] и ввести в использование понятие секретного ключа аутентификации, однозначно связанного с уникальным идентификатором субъекта. Далее будем обозначать множество ключей аутентификации символом $\mathbb{K}_a$.

Для использования асимметричной ключевой системы введём в рассмотрение множество ключей проверки кода аутентификации, которое будем обозначать символом $\mathbb{K}_c$, а также однонаправленную функцию $h : \mathbb{K}_a \to \mathbb{K}_c$, такую, что для любого $K_a \in \mathbb{K}_a$ выполнено условие $h(K_a) \in \mathbb{K}_c$. Под однонаправленной функцией h будем

подразумевать эффективно вычислимую функцию, для которой неизвестен эффективный алгоритм обращения [46, с. 79]. Для симметричной ключевой системы будем полагать, что выполнено равенство $\mathbb{K}_a = \mathbb{K}_c$, а h есть тривиальное отображение, не изменяющее значение своего аргумента.

Механизм связывания идентификатора субъекта с его ключом аутентификации зависит от схемы выработки ключей аутентификации:

— для асимметричных ключевых схем связывание происходит путём включения идентификатора и другой аутентифицирующей информации в состав сертификата открытого ключа участника протокола;
— для симметричных схем уникальные идентификаторы $ID_{A_1}, ID_{A_2}, \ldots ID_{A_r}$, определяемые для некоторого целого $r \geqslant 2$, используются при выработке общей ключевой информации для группы субъектов $A_1, A_2, \ldots, A_r$; примером такой схемы является ключевая система, регламентированная Р 1323565.028-2019.

Определение 6. Будем говорить, что криптографический протокол обеспечивает свойство аутентификации субъекта B, выполняемой субъектом A (свойство С1), если:

1) с субъектом B связан идентификатор ID_B;
2) для субъекта B определены ключ аутентификации субъекта $K_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $K_c \in \mathbb{K}_c$, однозначно связанные с идентификатором ID_B;
3) для некоторого натурального m определены функции выработки кода аутентификации $\texttt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\texttt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$, такие, что $\texttt{conf}\,(K_c, M, \texttt{mac}(K_a, M)) = \texttt{true}$ для любых значений $K_a, K_c, M \in \mathbb{V}_\infty^*$;
4) субъекту A известны идентификатор ID_B субъекта B и подтверждённое значение ключа проверки кода аутентификации K_c;
5) в состав протокола входит последовательность шагов, представленная на рис. 2; символом $\in_R$ обозначается выбор случайного элемента из заданного множества;

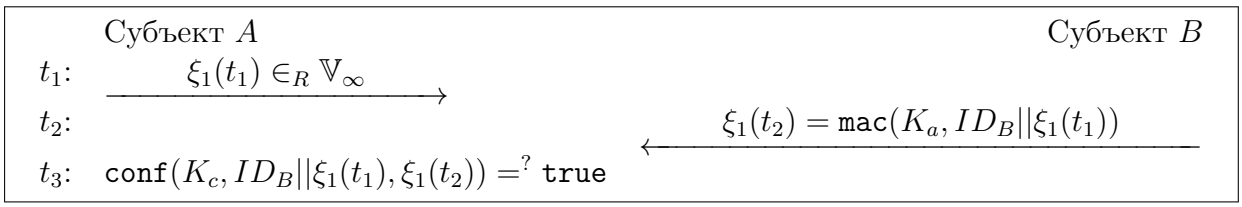

Рис. 2. Протокол аутентификации субъекта

6) истинно условие проверки на шаге t_3.

Данное определение согласуется как с симметричными, так и с асимметричными ключевыми системами — в качестве функции mac могут быть использованы как алгоритмы выработки имитовставки, так и алгоритмы выработки электронной подписи. Для общности изложения далее будем считать, что mac есть отображение с конечным числом аргументов, существенно зависящее от ключа аутентификации K_a, идентификатора субъекта B и случайного значения $\xi_1(t_1)$ и удовлетворяющее требованиям, предъявляемым к ключевым криптографическим функциям хеширования [46].

Начальное множество состояний субъекта A, выполняющего процесс аутентификации субъекта B, может быть определено следующим образом:

$$A(t_0) = \{a_1(t_0) = (K_c, \texttt{true}),\ a_2(t_0) = (ID_B, \texttt{false}),\ a_3 = (\varnothing, \texttt{false})\}.$$

Здесь ячейка a_1 соответствует ключу проверки кода аутентификации субъекта B, ячейка a_2 — идентификатору субъекта B, ячейка a_3 — вырабатываемому субъектом A случайному значению, используемому при проверке кода аутентификации. В следующие моменты времени состояние субъекта A имеет вид

$$\begin{array}{lll} A(t_1) = \{a_1(t_1) = (K_c, \texttt{true}), & a_2(t_1) = (ID_B, \texttt{false}), & a_3(t_1) = (\xi_1(t_1), \texttt{true})\}, \\ A(t_2) = \{a_1(t_2) = (K_c, \texttt{true}), & a_2(t_2) = (ID_B, \sigma), & a_3(t_2) = (\xi_1(t_1), \texttt{true})\}, \end{array}$$

где $\sigma = \texttt{conf}(K_c, ID_B||\xi_1(t_1), \xi_1(t_2))$, или может быть описано следующими нетривиальными функциями перехода:

$$\alpha_3(t_1) = \xi_1(t_1),\ \sigma_3(t_1) = \texttt{true},\ \sigma_2(t_2) = \texttt{conf}(\alpha_1(t_2), \alpha_2(t_2)||\alpha_3(t_2), \xi_1(t_2)).$$

Сделаем несколько замечаний к определению 6:

1) В рассмотренном протоколе (см. рис. 2) ключ проверки кода аутентификации K_c является исходной ключевой информацией для субъекта A. Поскольку именно эта информация обеспечивает аутентификацию субъекта B, её значение должно быть подтверждено до начала выполнения протокола, например, с помощью организационно-технических мер, удостоверяющего центра или в рамках другого протокола.
2) Из определения следует, что свойство аутентификации субъекта выполнено только для протоколов, включающих в себя взаимодействие субъектов (отправку и получение сообщений). Транспортные протоколы, предусматривающие только отправку сообщений от одного субъекта к другому, данному свойству не удовлетворяют. Вместе с тем использование в таких протоколах ключевой информации, владелец которой аутентифицирован ранее иным способом, позволяет говорить о наследовании свойства С 1.
3) Включение идентификатора ID_B в состав сообщения $\xi_1(t_2)$=$\texttt{mac}(K_a, ID_B||\xi_1(t_1))$ является принципиальным при использовании симметричной ключевой системы. В случае исключения идентификатора ID_B не представляется возможным предъявить алгоритмический способ различения того, кто из субъектов взаимодействия является автором пары сообщений $(\xi_1(t_1), \texttt{mac}(K_a, \xi_1(t_1)))$ (это следует из совпадения ключей аутентификации у обоих субъектов). Для асимметричной ключевой системы исключение идентификатора ID_B не является критичным, поскольку субъекты имеют различные ключи аутентификации.

Отметим, что на настоящий момент в Российской Федерации действует только морально устаревший стандарт ГОСТ Р ИСО/МЭК 9594-8-98 [47], регламентирующий процедуры аутентификации с использованием фактора знания секретного ключа. При этом определению 6 соответствует лишь часть процедур «строгой» аутентификации из [47], см. разд. 10. Современные стандарты серии ISO/IEC 9798 (части 1–6) в большинстве своём соответствуют определению 6. Протоколы из ISO/IEC 9798-5:2009 и ISO/IEC 9798-6:2010 могут формально не соответствовать этому определению и в случае необходимости их применения на территории Российской Федерации должны пройти дополнительное исследование на соответствие рассматриваемой модели.

2.2. Свойство целостности сообщений

В большинстве некриптографических протоколов для защиты от случайных искажений данные передаются вместе со своими кодами целостности, выработанными с помощью сжимающих отображений, таких, как Fletcher16 [48], CRC32 [49] и т. п., а также бесключевых криптографических функций хэширования, например функции «Стрибог» [50]. Используемая нами модель нарушителя делает применение таких функций бесполезным для защиты от преднамеренных искажений, а для обеспечения целостности, так же как и в п. 2.1, приходится использовать сжимающие преобразования, зависящие от секретного ключа.

Определение 7. Будем говорить, что криптографический протокол обеспечивает свойство целостности сообщения $M \in \mathbb{V}_\infty^*$, отправляемого субъектом B субъекту A (свойство С 3), если:

1) для субъекта B определены ключ аутентификации субъекта $K_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $K_c \in \mathbb{K}_c$;
2) субъекту A известно подтверждённое значение ключа проверки кода аутентификации субъекта K_c;
3) для некоторого натурального m определены функции выработки кода аутентификации $\mathtt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\mathtt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$, такие, что $\mathtt{conf}\,(K_c, M, \mathtt{mac}(K_a, M)) = \mathtt{true}$ для любых значений $K_a, K_c \in \mathbb{K}$, $M \in \mathbb{V}_\infty^*$;
4) в состав протокола входит последовательность шагов, представленная на рис. 3;

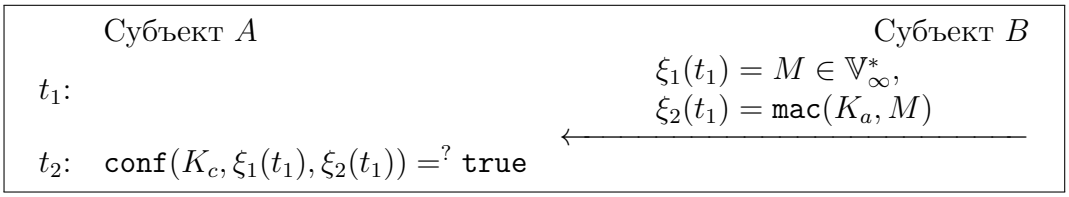

Рис. 3. Протокол подтверждения целостности сообщения

5) истинно условие проверки на шаге t_2.

Начальное множество состояний субъекта A, выполняющего процесс получения сообщений от субъекта B, может быть определено следующим образом:

$$A(t_0) = \{a_1(t_0) = (K_c, \mathtt{true}),\ a_2(t_0) = (\varnothing, \mathtt{false}),\ a_3 = (\varnothing, \mathtt{false})\}.$$

Здесь ячейка a_1 соответствует ключу проверки кода аутентификации субъекта B, ячейка a_2 — получаемому из канала связи сообщению, ячейка a_3 — коду целостности получаемого сообщения. В следующий момент времени состояние субъекта A имеет вид

$$A(t_1) = \{a_1(t_1) = (K_c, \mathtt{true}),\ a_2(t_1) = (\xi_1(t), \sigma),\ a_3(t_1) = (\xi_2(t_1), \sigma)\},$$

где $\sigma = \mathtt{conf}(K_c, \xi_1(t_1), \xi_2(t_1))$.

Отметим, что, как и в случае аутентификации субъекта, выполнение свойства целостности существенным образом зависит от того, подтверждена ли исходная ключевая информация — ключ проверки кода целостности K_c.

2.3. Свойство аутентификации сообщения

Свойство аутентификации сообщения (свойство С 2) является производным и следует из комбинации свойств аутентификации субъекта (свойство С 1, п. 2.1) и целостности передаваемых сообщений (свойство С 3, п. 2.2).

Определение 8. Криптографический протокол обеспечивает выполнение свойства аутентификации субъектом A сообщения $M \in \mathbb{V}_\infty^*$, отправленного субъектом B (свойство С 2), если в условиях определений 6 и 7 имеет место:

1) протокол содержит последовательность шагов, приведённую на рис. 4;

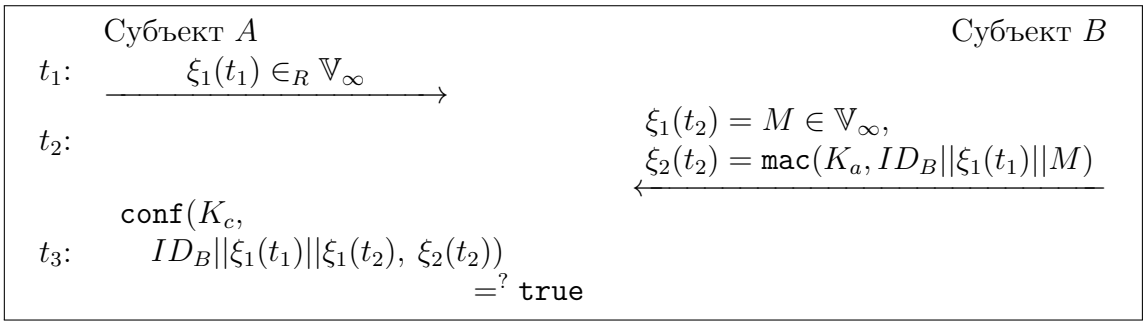

Рис. 4. Протокол аутентификации сообщения

2) истинно условие проверки на шаге t_3.

В данном определении мы совместили процедуры проверки кода аутентификации, выполняемые для проверки свойств аутентификации субъекта B и целостности сообщения M.

Следует отметить, что для выполнения свойства аутентификации нескольких отправляемых субъектом B сообщений (с целью снижения объёма передаваемой в канал связи информации) сообщение $\xi_1(t_1)$ может отправляться единожды, а значение $ID_B||\xi_1(t_1)$ заменяться на результат применения некоторого сжимающего отображения, известного субъектам A и B.

2.4. Свойство защиты от навязывания параметров безопасности

Свойство защиты от навязывания параметров безопасности (С 15) также является производным и основано на выполнении следующих свойств:

— аутентификации субъекта (свойство С 1),
— целостности сообщений (свойство С 3) и, как следствие,
— аутентификации сообщения (свойство С 2).

Достаточно часто в рамках одной сессии протокола между субъектами взаимодействия осуществляется согласование криптографических параметров, прямо или косвенно влияющих на безопасность передаваемых данных. В качестве таких параметров могут выступать алгоритмы блочного шифрования, функции хеширования, параметры циклической абелевой группы, в которых производится выработка производных ключей и т. п. Вмешательство нарушителя в процесс согласования параметров может привести к компрометации вырабатываемой в ходе выполнения протокола ключевой информации. Наиболее ярким примером являются атаки на понижение версии протокола TLS 1.2 [51].

Определение 9. Криптографический протокол обеспечивает выполнение свойства защиты от навязывания параметров безопасности $P \in \mathbb{V}_\infty$ субъекту A (свойство С 15), если сообщение, в котором передаются параметры безопасности для субъекта A, удовлетворяет свойству аутентификации сообщения (свойство С 2).

На практике согласование параметров безопасности выполняется в режиме «запрос — ответ», т. е. субъект A формирует перечень доступных для него параметров, а субъект B выбирает из них те, что будут использоваться в дальнейшем. Интепретируя параметры безопасности как строки из $\mathbb{V}_\infty$, получаем, что субъект A направляет субъекту B множество строк $\{S_1, \ldots, S_r\}$, а субъект возвращает одну строку S_i, $i \in \{1, \ldots, r\}$, из полученного множества. Тогда можем считать, что протокол, обеспечивающий свойство защиты от навязывания параметров безопасности, должен содержать последовательность шагов, представленную на рис. 5.

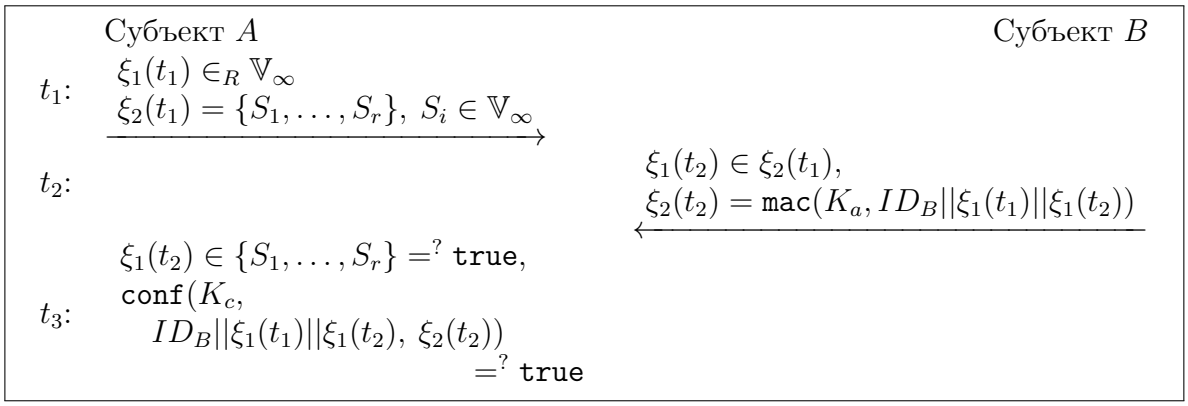

Рис. 5. Протокол защиты от навязывания параметров безопасности

В данном протоколе используется определяемая, как и ранее, исходная ключевая информация K_a, K_c. Субъект A сначала проверяет, что выбранный субъектом B параметр безопасности $\xi_1(t_2)$ принадлежит запрашиваемому множеству, а затем проверяет истинность кода аутентификации сообщения, содержащего $\xi_1(t_2)$. Отметим, что функции `mac` и `conf` должны быть зафиксированы до начала согласования параметров безопасности.

Примером описанного механизма может служить этап установления соединения в протоколе TLS 1.3, в котором все критически важные параметры передаются при помощи сообщения, подписанного электронной подписью сервера (субъекта B).

2.5. Свойства подтверждения и аутентификации ключа

Как мы видели ранее, для обеспечения свойств безопасности необходимо использование подтверждённой исходной ключевой информации K_c. В случае асимметричной ключевой системы подтверждение ключа K_c производится путём проверки электронной подписи удостоверяющего центра, выдавшего сертификат ключа K_c. В случае симметричной ключевой системы подтверждение обеспечивается организационно-штатными мерами доставки ключевой информации до субъектов взаимодействия.

Для производной ключевой информации, вырабатываемой в рамках криптографического протокола, требуется обеспечить подтверждение непосредственно в ходе взаимодействия субъектов.

Пусть $K_A \in \mathbb{K}_a$ — значение, выработанное субъектом A, а $K_B \in \mathbb{K}_a$ — значение, выработанное субъектом B. Теперь субъект A должен получить подтверждение того, что

значение K_A совпадает со значением K_B, т. е. проверить, что для некоторой функции f выполнено равенство

$$f(K_A, M) =^? f(K_B, M), \quad M \in \mathbb{V}_\infty,$$

в котором правая часть вычислена субъектом B, а левая часть — субъектом A. В качестве функции f может выступать, например, режим блочного шифрования или алгоритм выработки имитовставки.

Наиболее простой протокол, реализующий подтверждение субъектом A ключа K_A, изображён на рис. 6.

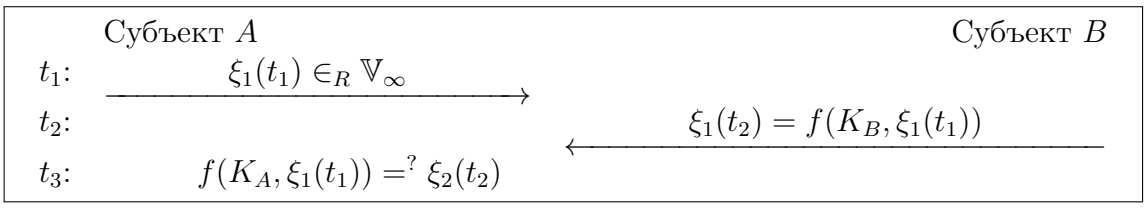

Рис. 6. Простой протокол подтверждения ключа

В данном протоколе субъект B с использованием подтверждаемого значения K_B преобразует случайное сообщение $\xi_1(t_1)$, а субъект A проверяет корректность результата преобразования с помощью подтверждаемого значения K_A. Сделаем ряд замечаний к протоколу рис. 6:

— Легко видеть, что сообщение $\xi_1(t_2)$ не удовлетворяет свойству аутентификации сообщений. Субъект A получает подтверждение, что он вычислил значение K_A правильно, но того, кто ему это подтверждение направил, субъект A идентифицировать не может. Тем самым у нарушителя появляется потенциальная возможность навязать субъекту A ложное значение ключа K_A.

— Другим недостатком предложенного протокола является возможность накопления нарушителем пар открытый/шифрованный текст $(\xi_1(t_1), f(K_B, \xi_1(t_1)))$ и их использование в дальнейшем для реализации алгоритмических методов определения секретного значения K_B либо для последующего навязывания ложных, но корректно зашифрованных значений. Это приводит к необходимости применять на этапе подтверждения ключа преобразование f, отличное от того, что будет использовано при взаимодействии субъектов.

Для учёта второго замечания можно модифицировать протокол рис. 6 и обмениваться только зашифрованными с помощью преобразования f сообщениями. Пример такой модификации приведён на рис. 7.

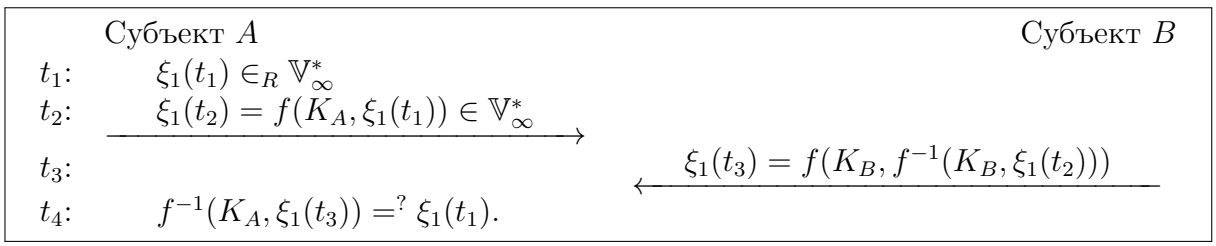

Рис. 7. Уязвимый протокол подтверждения ключа

В данном протоколе нарушитель не может накапливать пары открытый/шифрованный текст $(\xi_1(t_1), f(K_A, \xi_1(t_1)))$, поскольку значение $\xi_1(t_1)$ не передаётся субъектом A в канал связи. Предполагается также, что субъект B с помощью подтверждаемого ключа K_B сначала расшифровывает случайное сообщение $\xi_1(t_1)$, а потом повторно зашифровывает его (такую последовательность действий субъекта B можно назвать «расшифрование и зашифрование»).

Однако протокол уязвим, поскольку нарушитель может реализовать атаку «отражением сообщений» — перехватить сообщение $\xi_1(t_2) = f(K_A, \xi_1(t_1))$ и направить его обратно субъекту A вместо сообщения $\xi_1(t_3)$. После этого условие проверки на шаге t_4 всегда будет истинно, поскольку $f^{-1}(K_A, f(K_A, \xi_1(t_1))) = \xi_1(t_1)$.

Реализация атаки «отражением сообщений» становится возможной в силу следующих причин:

1) Построим формальную модель состояний субъекта A. Начальное множество состояний может быть определено следующим образом:

$$A(t_0) = \{a_1(t_0) = (K_A, \texttt{false}),\ a_2(t_0) = (\varnothing, \texttt{false})\}.$$

Здесь ячейка a_1 соответствует подтверждаемому ключу K_A, а ячейка a_2 — вырабатываемому случайному сообщению. Тогда последовательность состояний субъекта A описывается следующими нетривиальными функциями:

$$\alpha_2(t_1) = \xi_1(t_1), \quad \sigma_2(t_1) = \texttt{true}, \quad \sigma_1(t_4) = (\texttt{f}(\alpha_1(t_3), \xi_1(t_3)) =^{?} \alpha_2(t_3)).$$

Легко видеть, что содержимое ячейки a_1 подтверждается значением функции $f(\alpha_1(t_3), \xi_1(t_3)) =^{?} \alpha_2(t_3)$, не зависящим от какой-либо исходной ключевой информации.

2) С точки зрения субъекта A значение $\xi_1(t_3)$ рассматривается как реализация некоторой случайной величины. При этом ожидается, что вероятность угадывания нарушителем случайного значения $\xi_1(t_3)$, такого, что $\sigma_1(t_4)$ примет истинное значение, будет минимальной. Вместе с тем значение $\xi_1(t_3) = \xi_1(t_2)$ передаётся субъектом A в открытом виде. Это позволяет нарушителю перехватить его, отправить обратно субъекту A и с вероятностью 1 быть уверенным в том, что $\sigma_1(t_4) = \texttt{true}$.

Протокол рис. 7 иллюстрирует сделанное ранее высказывание (см. примечание к определению 5 на с. 46) о том, что подтверждение всех ячеек состояния субъекта является лишь необходимым условием безопасности протокола. Дополнительно должны рассматриваться вероятности подделки поступающих из канала связи значений, а также, в общем случае, и трудоёмкости алгоритмов подделки.

Защитой от атаки «отражением сообщений» является применение некоторого известного субъектам A и B преобразования h к неизвестному для нарушителя сообщению $\xi_1(t_1)$, т. е. вычисление равенства

$$\xi_1(t_3) = f(K_B, h(f^{-1}(K_B, \xi_1(t_2))))$$

(такую последовательность действий субъекта B можно назвать «расшифрование, преобразование и зашифрование»).

Если преобразование h отлично от преобразования f, является однонаправленным для нарушителя и не позволяет ему по значению $\xi_1(t_3) = h(x)$ определить значение аргумента x, то повторное применение преобразования f представляется излишним. В качестве однонаправленного преобразования h может выступать, например, бесключевая функция хеширования.

Определение 10. Будем говорить, что криптографический протокол обеспечивает для субъекта A свойство подтверждения факта обладания субъектом B ключом $K_B \in \mathbb{K}_a$ (свойство С 10), если:

1) субъект A обладает ключом $K_A \in \mathbb{K}_a$, для которого подтверждается выполнение равенства $K_A = K_B$;
2) определена зависящая от секретного ключа функция $f : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, для которой определена обратная функция $f^{-1} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, такая, что для любого сообщения $M \in \mathbb{V}_\infty$ равенство
$$f^{-1}(K_A, f(K_B, M)) = M$$
справедливо, когда $K_A, K_B \in \mathbb{K}_a$ и $K_A = K_B$;
3) для некоторого $m \in \mathbb{N}$ определена однонаправленная функция $h : \mathbb{V}_\infty \to \mathbb{V}_m$;
4) в состав протокола входит последовательность шагов, представленная на рис. 8;

	Субъект A		Субъект B
t_1:	$\xi_1(t_1) \in_R \mathbb{V}_\infty^*$		
t_2:	$\xi_1(t_2) = f(K_A, \xi_1(t_1)) \in \mathbb{V}_\infty^*$	$\xrightarrow{\qquad\qquad}$	
t_3:		$\xleftarrow{\qquad\qquad}$	$\xi_1(t_3) = h(f^{-1}(K_B, \xi_1(t_2)))$
t_4:	$th(\xi_1(t_1)) =^? \xi_1(t_3)$		

Рис. 8. Протокол подтверждения ключа

5) истинно условие проверки на шаге t_4.

Следует отметить, что условие существования (эффективно вычислимого субъектом B) обратного преобразования f^{-1} является необходимым, так как в противном случае вычисление сообщения $\xi_1(t_1) = f^{-1}(K, \xi_1(t_2))$ невозможно.

Используя в качестве преобразования h функцию вычисления кода аутентификации `mac`, можно добиться зависимости значения $\sigma_1(t_4)$ от исходной ключевой информации. В этом случае необходимо использовать равенство
$$\xi_1(t_3) = \texttt{mac}(K_a, f^{-1}(K_B, \xi_1(t_2)))).$$

Учитывая, что сообщение $\xi_1(t_3)$ должно удовлетворять свойству аутентификации сообщений (свойство С 2), дадим ещё одно определение.

Определение 11. Будем говорить, что криптографический протокол обеспечивает для субъекта A свойство аутентификации принадлежащего субъекту B ключа $K_B \in \mathbb{K}_a$ (свойство С 9), если:

1) с субъектом B связан идентификатор ID_B;
2) для субъекта B определены ключ аутентификации субъекта $K_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $K_c \in \mathbb{K}_c$, однозначно связанные с идентификатором ID_B;
3) субъекту A известны идентификатор ID_B субъекта B и подтверждённое значение ключа проверки кода аутентификации K_c;
4) субъект A обладает ключом $K_A \in \mathbb{K}_a$, для которого подтверждается выполнение равенства $K_A = K_B$;

5) для некоторого натурального m определены функции выработки кода аутентификации $\mathtt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\mathtt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$, такие, что

$$\mathtt{conf}\,(K_c, M, \mathtt{mac}(K_a, M)) = \mathtt{true},$$

для любых значений $K_a, K_c, M \in \mathbb{V}_\infty^*$;

6) определена зависящая от секретного ключа функция $f : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, для которой определена обратная функция $f^{-1} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, такая, что для любого сообщения $M \in \mathbb{V}_\infty$ равенство

$$f^{-1}(K_A, f(K_B, M)) = M$$

справедливо, когда $K_A, K_B \in \mathbb{K}_a$ и $K_A = K_B$;

7) в состав протокола входит последовательность шагов, представленная на рис. 9;

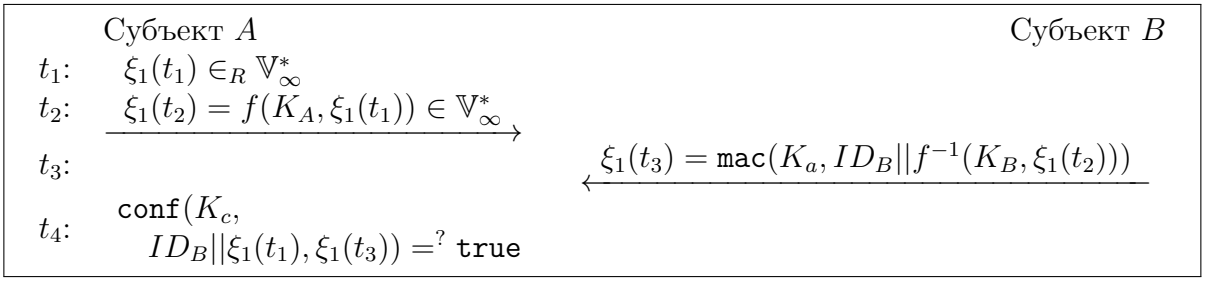

Рис. 9. Протокол подтверждения ключа

8) истинно условие проверки на шаге t_4.

Отметим, что формально следуя определению 8, можно было бы определить множество отправляемых субъектом B сообщений следующим образом:

$$\xleftarrow{\begin{array}{c}\xi_1(t_3) = \mathtt{mac}(K_B, \mathtt{f}^{-1}(K_B, \xi_1(t_2))),\\ \xi_2(t_3) = \mathtt{mac}(K_a, ID_B||\xi_1(t_2)||\xi_1(t_3)).\end{array}}$$

Предложенный на рис. 8 вариант достигает той же цели с меньшей длиной данных, передаваемых в канал связи.

Отметим, что если функции f, mac и conf согласуются в ходе выполнения протокола, то для протокола должно быть выполнено свойство защиты от навязывания параметров (свойство С 15).

2.6. Свойство конфиденциальности ключа

Свойство конфиденциальности ключа является базовым и применяется как к исходной ключевой информации, так и к сессионным (производным) ключам, вырабатываемым в ходе выполнения протокола. Формальное определение свойства конфиденциальности ключа тесно связано с показателями эффективности, рассматриваемыми далее в п. 3.2.

Определение 12. Пусть заданы $K_a \in \mathbb{K}_a$ — исходная ключевая информация, $h : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ — некоторая однонаправленная функция и множество пар $(\xi_{i_k}(t_k), h(K_a, \xi_{i_k}(t_k)))$, перехваченных нарушителем в ходе выполнения одной или нескольких сессий протокола.

Будем говорить, что протокол обеспечивает конфиденциальность исходной ключевой информации (свойство С 8), если нарушитель не может определить значение K_a с вероятностью, большей чем π_0 (см. определение 1), и трудоёмкостью, не превосходящей некоторое значение Q_0.

Легко видеть, что данное определение может быть расширено на случай нескольких однонаправленых функций $h_1, h_2, \ldots$

Отличие производной ключевой информации от исходной заключается в том, что она вырабатывается в ходе выполнения протокола. В случае, когда выработка происходит с использованием значений, передаваемых между субъектами взаимодействия, протокол должен реализовывать механизмы защиты передаваемых значений от подделки и навязывания нарушителем.

Определение 13. Пусть заданы $K_a \in \mathbb{K}_a$ — исходная ключевая информация, отображение $\mathtt{var} : \mathbb{K}_a \times \mathbb{V}_\infty^* \times \cdots \times \mathbb{V}_\infty^* \to \mathbb{K}_a$, используемое для выработки производной ключевой информации, и $h : \mathbb{K} \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ — некоторая однонаправленная функция.

Будем говорить, что протокол обеспечивает конфиденциальность производной ключевой информации K_A (свойство С 8), если выполнены следующие условия:

1) величина K_A (в общем виде) определяется равенством

$$K_A = \mathtt{var}(K_a, \xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}}),\ \beta_{j_1}, \ldots, \beta_{j_r}), \tag{1}$$

   где $\beta_{j_1}, \ldots, \beta_{j_r}$ определены равенствами $\xi_{j_s}(t_{k_{j_s}}) = h(\beta_{j_s})$, $s = 1, \ldots, r$, а величины $\xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$ и $\xi_{j_1}(t_{k_{j_1}}), \ldots, \xi_{j_r}(t_{k_{j_r}})$ передаются между субъектами взаимодействия в ходе выполнения протокола (они могут перехватываться нарушителем);
2) в равенстве (1) либо переменные $K_a, \xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$, либо переменные $\beta_{j_1}, \ldots, \beta_{j_r}$ могут являться несущественными;
3) если переменные $\xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$ являются существенными, то они должны передаваться в составе сообщений, для которых выполнено свойство аутентификации сообщений (свойство С 2, п. 2.3);
4) если переменные $\beta_{j_1}, \ldots, \beta_{j_r}$ являются существенными, то они не могут быть определены нарушителем с вероятностью, большей чем величина π_0, и трудоёмкостью, не превосходящей некоторое значение Q_0;
5) после выработки производной ключевой информации K_A она должна использоваться таким образом, чтобы удовлетворять определению 12.

2.7. Свойство конфиденциальности

Согласно ГОСТ Р ИСО/МЭК 27033-1:2011, угроза нарушения конфиденциальности передаваемой информации является одной из основных угроз при обеспечении безопасности сетей связи. Однако свойство конфиденциальности (свойство С 16) не является базовым и выполняется только при выполнении совокупности рассмотренных ранее свойств.

Во-первых, используемый для шифрования информации ключ, являющийся, как правило, производной ключевой информацией, должен быть неизвестен нарушителю, т. е. удовлетворять свойству конфиденциальности ключа (свойство С 8).

Во-вторых, этот ключ должен удовлетворять свойству аутентификации ключа (свойство С 9). Это позволит получающему сообщения субъекту быть уверенным в том, что он не только получает сообщения, зашифрованные на том самом ключе, который

используется для их расшифрования, но и в том, что отправителем этих сообщений является аутентифицированный субъект взаимодействия.

В-третьих, для зашифрования информации, передаваемой в информационных системах, попадающих под действие нормативного регулирования, допускается применять только алгоритмы, входящие в национальную систему стандартизации Российской Федерации. Перечень допустимых алгоритмов шифрования определяется согласно ГОСТ Р 34.12-2015, ГОСТ Р 34.13-2015, а также ряда рекомендаций по стандартизации.

Схема зависимостей свойства конфиденциальности приведена на рис. 10.

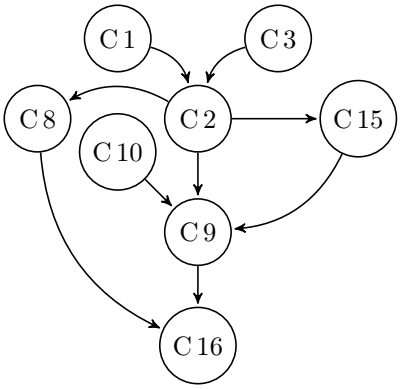

Рис. 10. Схема зависимостей свойства конфиденциальности

Далее рассмотрим ряд свойств, нарушение которых может привести к появлению негативных эффектов, не предполагаемых спецификацией криптографического протокола.

2.8. Свойство целостности множества состояний

С целью защиты от атак, использующих для компрометации одной сессии протокола данные, перехваченные в ходе выполнения другой сессии, рассмотрим свойство целостности множества состояний (свойство С 27). Пусть

$$A(t_0) = \{a_1, \ldots, a_{n(A)} : \; a_i = \big(\alpha_i(t_0), \sigma_i(t_k)\big)\}$$

— начальное состояние субъекта A. Будем считать, что для некоторого натурального $n_1(A)$, такого, что $1 \leqslant n_1(A) < n(A)$, ячейки $a_1, \ldots, a_{n_1(A)}$ содержат исходную ключевую информацию, а также любые другие значения, подтверждённые до начала выполнения протокола, т. е.

$$\sigma_i(t_0) = \texttt{true}, \quad i = 1, \ldots, n_1(A).$$

Также будем считать, что для некоторого натурального $n_2(A)$, такого, что $n_1(A) < n_2(A) < n(A)$, ячейки $a_{n_1(A)+1}, \ldots, a_{n_2(A)}$ содержат случайные значения, вырабатываемые субъектом A с использованием генератора случайных чисел (ГСЧ), т. е. найдутся такие временные метки t_{k_i}, что

$$\sigma_i(t_{k_i}) = \texttt{true}, \quad i = n_1(A) + 1, \ldots, n_2(A).$$

Определение 14. Будем говорить, что криптографический протокол удовлетворяет свойству целостности множества состояний (свойство С 27) для субъекта A, если найдется такая временная метка t_{k_0}, что для всех $t_k \geqslant t_{k_0}$ выполнено

$$\sigma_i(t_k) = \mathtt{conf}(a_1(t_{k-1}), \ldots, a_{n_2(A)}(t_{k-1}), \ldots), \quad i = n_2(A) + 1, \ldots, n(A), \tag{2}$$

и зависимость от $a_1(t_{k-1})$, …, $a_{n_2(A)}(t_{k-1})$ является существенной.

Будем говорить, что криптографический протокол удовлетворяет строгому свойству целостности множества состояний для субъекта A, если одновременно с условием (2) выполнено

$$\alpha_i(t_k) = \mathtt{var}(\ldots, \alpha_{n_1(A)+1}(t_{k-1}), \ldots, \alpha_{n_2(A)}(t_{k-1}), \ldots), \quad i = n_2(A) + 1, \ldots, n(A), \tag{3}$$

и зависимость от $\alpha_{n_1(A)+1}(t_{k-1})$, …, $\alpha_{n_2(A)}(t_{k-1})$ является существенной.

Если криптографический протокол удовлетворяет определению 14, то субъект A может удостовериться в том, что каждая из ячеек памяти его состояния, вырабатываемая в ходе выполнения протокола, подтверждается с использованием значений, которые не могут быть навязаны нарушителем.

Существенная зависимость от значений $\alpha_{n_1(A)+1}, \ldots, \alpha_{n_2(A)}$, вырабатываемых с использованием ГСЧ, позволяет говорить о том, что данные значения выработаны непосредственно в ходе выполнения протокола, т. е. в реальном времени, и не могут быть продублированы в ходе выполнения другой сессии протокола. Невозможность дублирования случайных значений должна обеспечиваться используемым ГСЧ.

2.9. Свойство защищённости от KCI-атак

Рассмотрим свойство защищённости от KCI-атак — Key Compromise Impersonation attack (свойство С 22). Данные атаки реализуются в случае компрометации исходной ключевой информации (долговременного ключа) одного из легальных субъектов, или в случае определения нарушителем исходной ключевой информации (нарушения свойства С 8, п. 2.6).

В качестве примера такой атаки рассмотрим протокол MTI(C0) [19] и построим KCI-атаку для него.

Пусть q — нечётное простое число, $G = \langle g \rangle$ — циклическая абелева группа, порождаемая элементом g порядка q. Будем считать, что в группе G решение задачи дискретного логарифмирования имеет высокую трудоёмкость.

Субъекты A и B обладают парами асимметричных ключей (закрытый и открытый): $a \in \mathbb{F}_q^*$, $K_{cA} = g^a \in G$ и $b \in \mathbb{F}_q^*$, $K_{cB} = g^b \in G$. Будем считать, что открытые ключи K_{cA}, K_{cB} известны обоим субъектам взаимодействия, а их значения подтверждены до начала выполнения протокола. Схема работы протокола MTI(C0) представлена на рис. 11.

Рассмотрим реализацию KCI-атаки на протокол MTI(C0) в рамках предположения, что нарушитель C знает закрытый ключ a участника A и открытые ключи K_{cA} и K_{cB}. Нарушитель C пытается выдать себя за субъекта B перед субъектом A (рис. 12).

При реализации атаки нарушитель C может сформировать сообщение $\xi_1(t_4)$ от лица субъекта B (при условии знания ключа a) таким образом, что субъект A ничего не заподозрит и будет думать, что ключ выработан с субъектом B, а на самом деле он будет выработан с нарушителем C.

В качестве основных методов противодействия KCI-атакам можно выделить следующие:

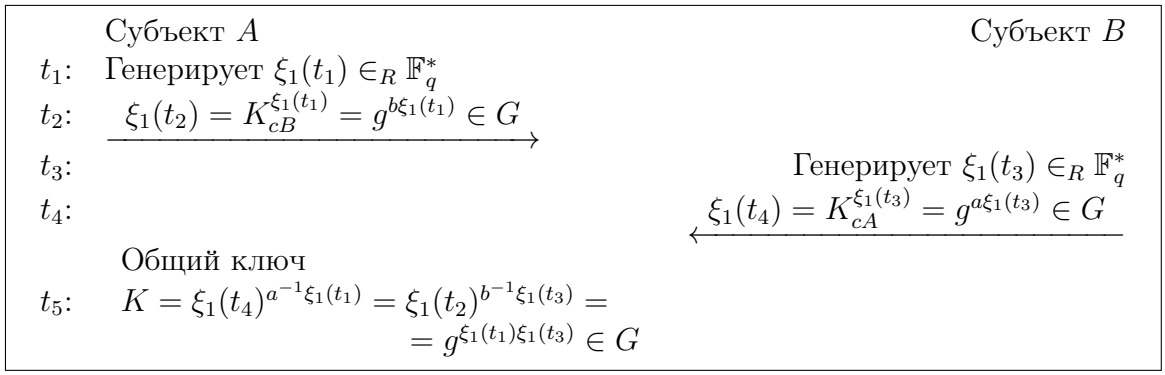

Рис. 11. Протокол MTI(C0)

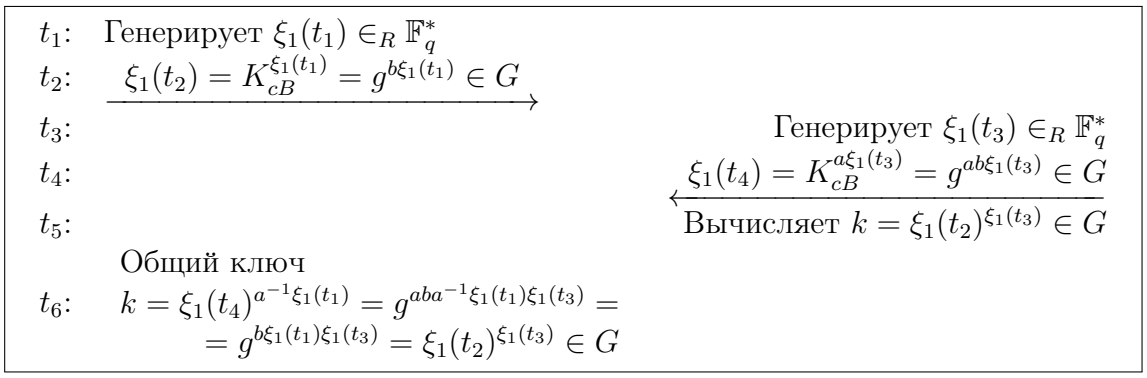

Рис. 12. KCI-атака на протокол MTI(C0)

— уникальность сессионных ключей — ключи для каждой сессии протокола должны генерироваться независимо;
— запрет на использование ключей аутентификации при формировании сессионных ключей (см., например, равенство (3));
— использование производных ключей, сформированных субъектами в процессе выполнения протокола независимо друг от друга, при выработке сессионного ключа;
— обязательная аутентификация субъекта, с которым производится взаимодействие, а также аутентификация выработанного сессионного ключа.

Поскольку KCI-атаки нацелены на эксплуатацию недостатков ключевой системы протокола и механизмов аутентификации, то можно говорить о защищённости протокола от данного класса атак только в случае выполнения следующих свойств: C 1 (аутентификация субъекта), C 9 (аутентификация ключа), C 10 (подтверждение ключа), C 12 (стойкость при компрометации ключа аутентификации), C 13 (формирование новых ключей) и C 14 (защита от навязывания ключевых значений).

3. Определение показателей эффективности защиты информации

В настоящее время в Российской Федерации принято оценивать стойкость средств криптографической защиты информации с точки зрения «практической стойкости», т. е. путём оценки трудоёмкости известных аналитику методов решения математических задач, решение которых приводит к компрометации используемых криптографических преобразований и алгоритмов. Изложение принятой методологии оценки стойкости с разной долей детализации может быть найдено в работах [20, 21, 52, 53].

При этом минимальная трудоёмкость и вероятность успешного решения рассматриваемых задач выступают в качестве показателей эффективности защиты. Представляется естественным распространить эти же показатели и на анализ криптографических протоколов.

В соответствии с определённой выше моделью нарушителя оценка возможности компрометации криптографического протокола может осуществляться при помощи следующих подходов:

1) При помощи «пассивных» атак, т. е. перехвата, накопления и последующего анализа перехваченной информации. В рамках применяемой в работе модели нарушителя криптографического протокола такие атаки сводятся к обращению однонаправленных функций, т. е. к решению сложных математических задач. Для каждой из таких задач рассматривается некоторое множество алгоритмов, находящих решение задачи с вероятностью π и трудоёмкостью Q. Отбрасывая алгоритмы с ничтожной вероятностью успеха, меньшей чем некоторое заранее фиксированное значение π_0, мы можем выбрать алгоритм с наименьшей трудоёмкостью. Именно такой алгоритм и считается наилучшим алгоритмом компрометации криптографического протокола при проведении «пассивных» атак.
2) При помощи «активных» атак, сводящихся к навязыванию одному или нескольким субъектам ложных значений, поступающих из канала связи; цель такого навязывания состоит в том, чтобы заставить легитимного субъекта сделать ложный вывод о том, что значения одной или нескольких ячеек его памяти являются истинными (подтверждёнными).

   Навязываемые значения могут вычисляться нарушителем как случайным образом, так и с использованием методов, применяемых при реализации «пассивных» атак. В первом случае считаем, что трудоёмкость выработки навязываемых значений ничтожна и основную роль при анализе играет вероятность π принять ложное значение за истинное, при этом число попыток навязывания ограничено только временем действия исходной ключевой информации и спецификацией протокола (если спецификация содержит подобные ограничения). Если полученное после исследования значение вероятности π меньше, чем заранее фиксированное значение π_0, то отбрасываем такой способ компрометации протокола как маловероятный.

   Во втором случае, когда нарушитель вырабатывает навязываемые значения путём решения сложных математических задач, в качестве вероятности успеха π естественно принять величину вероятности успеха алгоритма, имеющего наименьшую трудоёмкость реализации Q.

Описанные подходы позволяют получать единообразные численные значения показателей эффективности защиты, в качестве которых будем использовать минимально допустимую вероятность успеха алгоритма компрометации криптографического протокола π_0 и минимальную трудоёмкость Q_0 алгоритма компрометации, имеющего вероятность успеха, большую или равную π_0.

Дадим более формальное описание сказанного. Обозначим $k = k_{\max} - 1$ и рассмотрим $\mathbb{V}_{n_j}(t_k)$, $j = 1, \ldots, l_k$, — конечные множества, задающие область определения случайных величин, принимающих на указанных множествах в момент времени t_k соответственно значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$. Тогда величина, определяемая равенством

$$\sigma_i(t_{k_{\max}}) = \mathtt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \in \mathbb{B},$$

в котором $a_j(t_k) = \{\alpha_j(t_k), \sigma_j(t_k)\}$, $j = 1, \ldots, n(A)$ — некоторые фиксированные значения из $\mathbb{V}^*_\infty \times \mathbb{B}$, такие, что для каждой существенной ячейки памяти выполнено условие $\sigma_j(t_k) = \mathtt{true}$, может рассматриваться как реализация случайной величины, принимающей два значения — истина или ложь.

Определим символом

$$\pi_{i,k_{\max}} = \begin{cases} 0, & \text{если } l_k = 0, \\ \mathsf{P}[\sigma_i(t_{k_{\max}}) = \mathtt{true}] & \text{иначе} \end{cases} \tag{4}$$

вероятность принять случайный вектор $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ в качестве значения, подтверждающего $\alpha_i(t_{k_{\max}})$ в момент времени $t_{k_{\max}}$. Будем говорить, что величина $\pi_{i,k_{\max}}$ определяет вероятность принять ложное значение $\alpha_i(t_{k_{\max}})$ за истинное.

Поскольку значение каждой из существенных переменных $a_i(t_k)$ должно быть подтверждено, для них также могут быть определены вероятности $\pi_{i,k}$ принять ложное значение $\alpha_i(t_k)$ за истинное. После этого мы аналогично должны определить вероятности $\pi_{i,k-1}$, $\pi_{i,k-2}$ и так далее.

Представляя множество зависимостей между переменными в качестве графа, мы можем расположить в его узлах значения $\pi_{i,k}$, $k = 1, \ldots, k_{\max}$, и задать величину

$$\pi = \max_{L_{i,j}} \left(1 - \prod_{\pi_{i,k} \in L_{i,j}} (1 - \pi_{i,k}) \right), \quad j \in \mathbb{N}, \tag{5}$$

где $L_{i,1}, L_{i,2}, \ldots$ — пути в графе, приводящие к значению ячейки $a_i(t_{k_{\max}})$.

Определение 15. Величину π, задаваемую равенством (5), будем называть вероятностью успешной компрометации криптографического протокола.

Выбор значений $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, определяющих вероятности $\pi_{i,k}$, может производиться нарушителем двумя способами: случайным образом или с использованием предварительных вычислений. Рассмотрим оба способа подробнее.

3.1. Случайное угадывание

Мы можем рассматривать принимаемые субъектом A величины $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ как случайные значения, для которых задано распределение вероятностей

$$\mathsf{P}[\xi_j(t_k) = v] = \frac{1}{|\mathbb{V}_{n_j}(t_k)|}, \quad j = 1, \ldots, l_k, \quad v \in \mathbb{V}_{n_j}(t_k).$$

Такая ситуация возникает, когда $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ вырабатываются субъектом B с использованием секретного ключа, не известного нарушителю. В этом случае нарушитель просто угадывает значения, выбирая их случайным образом из области определения. В этом случае вероятность

$$\mathsf{P}[\mathtt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) = \mathtt{true}] = \pi_{i,k}$$

определяет вероятность однократного навязывания значения $\alpha_i(t_k)$.

В качестве примера рассмотрим изображённый на рис. 2 протокол аутентификации. Для ложной аутентификации нарушителю нужно предъявить значение $\xi_1(t_2)$, такое, чтобы у субъекта A выполнялось равенство

$$\mathtt{conf}(K_c, \alpha_1(t_1)||\alpha_2(t_1), \xi_1(t_2)) = \mathtt{true}$$

при $\alpha_1(t_1) = ID_B$, $\alpha_2(t_1) = \xi_1(t_1)$ (значение $\xi_1(t_1)$ отправляется субъектом A в канал связи и доступно нарушителю).

Поскольку нарушителю неизвестен секретный ключ K_a, он выбирает значение кода аутентификации случайным образом. Если в качестве функции `mac` используется алгоритм выработки электронной подписи, регламентированный ГОСТ Р 34.10-2012, и используется эллиптическая кривая с порядком группы точек q бит, где q — нечётное простое число, то вероятность однократного нарушения свойства аутентификации может быть оценена величиной $\pi_{1,3} = (q-1)/2$ (множитель $1/2$ возникает из-за того, что в алгоритме выработки электронной подписи используется только x-координата точки эллиптической кривой и точки (x, y) и $(x, -y)$ дают одинаковое значение подписи).

Рассмотрим другой пример, возникающий при исследовании транспортных криптографических протоколов. Пусть субъект A принимает от субъекта B аутентифицируемые сообщения M_1, M_2, ... (свойство С 2, п. 2.3) в соответствии со схемой, изображённой на рис. 13.

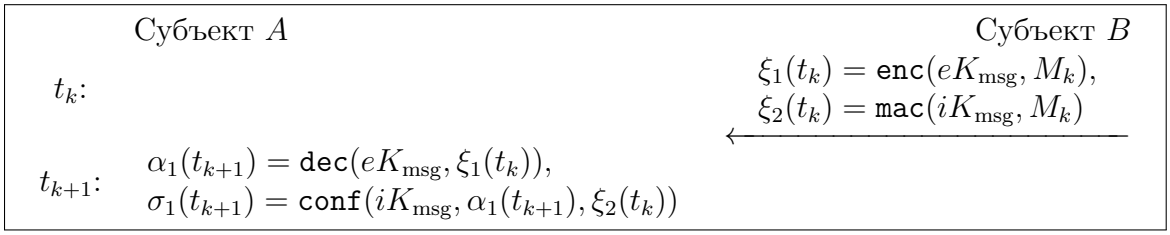

Рис. 13. Транспортный протокол приема сообщений

При этом мы предполагаем, что субъект B и производные ключи eK_{msg}, iK_{msg} были ранее аутентифицированы субъектом A в процессе выполнения протокола выработки общих ключей. Такая ситуация реализуется в большинстве современных протоколов, например TLS 1.3, IPSec, SP FIOT и т. п. Также будем считать, что для выработки кода аутентификации (значения $\xi_2(t_k)$) длиной n бит используется алгоритм, рекомендуемый отечественными стандартами или рекомендациями по стандартизации.

Применяя предложенный выше подход, мы можем оценить вероятность навязывания нарушителем субъекту одного ложного сообщения в течение заданного интервала времени. Обозначим символом V пропускную способность канала связи, по которому передаются сообщения (на практике могут использоваться значения 100 Мбайт/с, 1Гбит/с, 100 Гбит/с и т. п.). За время T (измеряемое в секундах) может быть передано не более VT бит информации, или не более $m = \lceil VT/l \rceil$ сообщений, где l — минимально возможная длина (в битах) сообщений $M_1, M_2, \ldots$ Тогда, согласно равенству (5), получаем, что для $m \leqslant n$ вероятность навязывания ложного сообщения равна

$$\pi = 1 - \left(1 - \frac{1}{2^n}\right)^m =$$

$$= \frac{m}{2^n} - \frac{m(m-1)}{2^{2n}} + \frac{m(m-1)(m-2)}{2^{3n}} + \ldots + \frac{1}{2^{mn}} = \sum_{i=1}^{m} \frac{(-1)^{i-1} m!}{i!(m-i)!2^{in}},$$

где 2^{-n} — вероятность случайного угадывания значения кода аутентификации. При $m \geqslant n$ считаем, что $\pi = 1$.

3.2. Применение вычислительных алгоритмов

Пусть, как и ранее, $k = k_{\max} - 1$ и

$$\sigma_i(t_{k_{\max}}) = \mathtt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \in \mathbb{B}.$$

Для подделки значений $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, получаемых субъектом A из канала связи, нарушитель может использовать подход, отличный от случайного выбора значений.

Обозначим символом $\Xi_{i,k_{\max}}$ множество передаваемых по каналу связи значений

$$\Xi_{i,k_{\max}} = \{\xi_1(t_1), \ldots, \xi_{l_1}(t_1),\ \xi_2(t_1), \ldots, \xi_{l_2}(t_2), \ldots, \xi_1(t_k), \ldots, \xi_{i-1}(t_k)\} \tag{6}$$

и будем считать, что для некоторого индекса $j \in [1, \ldots, l_k]$ значение $\xi_j(t_k)$ определяется субъектом B равенствами

$$\xi_j(t_k) = f(\Xi^{(1)}_{j,k}, B_{j,k}, \Gamma_{j,k}), \tag{7}$$

в которых:

— $\Xi^{(1)}_{j,k} \subseteq \Xi_{j,k}$ — множество переданных ранее по каналу связи значений, элементы которых известны нарушителю;

— $B_{j,k} = \{\beta_1, \ldots, \beta_{r_{j,k}}\}$, где $r_{j,k} \in \mathbb{N}$, — множество значений, удовлетворяющих равенствам

$$\xi_u(t_v) = h_j(\beta_j) \in \Xi_{i,k}$$

для некоторых $u, v \in \mathbb{N}$, $v \leqslant k$, и однонаправленных отображений $h_1, \ldots, h_{r_{j,k}}$ (в общем случае используемые при определении величин $\beta_1, \ldots, \beta_{r_{j,k}}$ значения $\xi_u(t_v)$ могут не принадлежать множеству $\Xi^{(1)}_{j,k}$);

— $\Gamma = \{\gamma_1, \ldots, \gamma_{s_k}\}$ — множество произвольно формируемых субъектом B значений, где s_k — целое неотрицательное число.

Отметим, что преобразование f может быть составным и включать выработку производной ключевой информации, используемой не только при получении значения $\xi_j(t_k)$, но и других значений, вычисляемых позднее.

В качестве примера рассмотрим изображённый на рис. 14 вариант протокола Диффи — Хеллмана, реализуемый в циклической абелевой группе $G = \langle g \rangle$, порождаемой элементом g порядка q, где q — нечётное простое число. Как и ранее, будем считать, что в группе G решение задачи дискретного логарифмирования имеет высокую трудоёмкость.

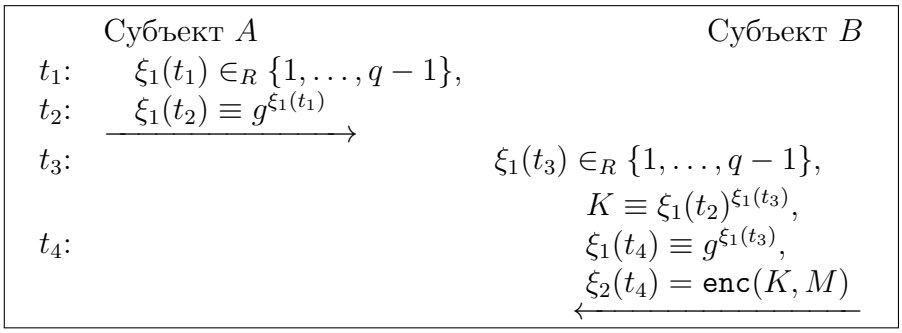

Рис. 14. Протокол Диффи — Хеллмана с передачей зашифрованного сообщения

В примере субъект A принимает от субъекта B сообщение M, зашифрованное на общем ключе K. Тогда значение $\xi_2(t_4)$, в соответствии с (7), может быть представлено

в виде

$$\xi_2(t_4) = f\left(\Xi_{2,4}^{(1)} = \{\xi_1(t_2)\},\ B_{2,4} = \{\xi_1(t_3)\},\ \Gamma_{2,4} = \{M\}\right),$$

где отображение f представляет собой процедуру зашифрования сообщения M, а однонаправленное отображение h_1 определяется сравнением

$$\xi_1(t_4) \equiv g^{\xi_1(t_3)} \pmod{p}$$

и представляет собой операцию возведения в степень в группе G (отметим, что протокол рис. 14 является иллюстрацией равенства (7) и не безопасен, поскольку общий ключ K может быть навязан субъекту A с помощью «атаки посередине», нарушено также свойство целостности передаваемого сообщения M, п. 2.2).

Вернёмся к (7). Нарушитель, которому известны отображения $f, h_1, \ldots, h_{r_k}$, может определить значения $\beta_j = h_j^{-1}(\xi_j(t_{l_j}))$ и навязать субъекту A ложные значения $\gamma_1, \ldots, \gamma_{s_k}$. Однонаправленные отображения $h_1, \ldots, h_{r_k}$, как правило, выбираются при синтезе протокола таким образом, чтобы вычисление обратного отображения являлось сложной математической задачей. К таким задачам могут быть отнесены задача определения секретного ключа алгоритма блочного шифрования, задача дискретного логарифмирования в группе точек эллиптической кривой, задача вычисления первого прообраза для функции хеширования и т. п.

Для каждой из перечисленных задач могут быть рассмотрены методы её решения, характеризуемые трудоёмкостью $Q_{j,k}$ и вероятностью успеха $p_{j,k}$, для которой выполнено неравенство $p_{j,k} \geqslant \pi_0$ (значение π_0 позволяет вывести из рассмотрения алгоритмы с ничтожной вероятностью успеха). Тогда вероятность навязывания ложных значений $\gamma_1, \ldots, \gamma_{s_k}$, или, другими словами, компрометации ячейки памяти $a_i(t_{k_{\max}})$, может быть определена равенством

$$\pi_{i,k_{\max}} = \prod_{j=1}^{r_k} p_{j,k}, \quad k = k_{\max} - 1$$

и трудоёмкостью

$$Q_{i,k_{\max}} = \sum_{j=1}^{r_k} Q_{j,k}.$$

Вероятность π успешной компрометации протокола в целом определяется с использованием равенства (5). При этом общая трудоёмкость компрометации протокола задаётся величиной

$$Q = \min_{L_{i,j}} \sum_{\pi_{i,k} \in L_{i,j}} Q_{i,k},$$

где $L_{i,1}, L_{i,2}, \ldots$ — пути в графе, приводящие к значению ячейки $a_i(t_{k_{\max}})$. Сделаем ряд замечаний:

1) Определённое равенством (6) множество $\Xi_{j,k}$ состоит из значений, передаваемых в ходе выполнения одной сессии выполнения протокола. Однако, согласно принятой модели нарушителя, для навязывания значений $\gamma_1, \ldots, \gamma_{s_k}$ нарушитель может использовать значения, передаваемые во всех сессиях, выполнявшихся до момента проведения атаки $t_{k_{\max}}$. Это необходимо учитывать при определении вероятности и трудоёмкости обращения значений однонаправленных функций.
2) Трудоёмкость компрометации протокола при случайном выборе нарушителем значений $\xi_1(t_k), \ldots, \xi_1(t_{l_k})$, см. п. 3.1, определяется длиной пути в графе, соответствующей максимальной вероятности компрометации протокола.

Теперь мы можем дать определение безопасного криптографического протокола. Будем считать, что допустимые значения показателей эффективности защиты π_0 и Q_0 заданы и, согласно рекомендациям Р 1323565.1.012-2017, определены действующими требованиями по безопасности для каждого класса средств защиты информации.

Определение 16. Будем говорить, что протокол обладает подтверждённым состоянием субъекта A в момент времени t_{k_A}, где $k_A \in \{1, \ldots, k_{\max}\}$, если значение всех ячеек памяти состояния $A(t_{k_A})$ является подтверждённым явно или косвенно, т. е.

$$\sigma_i(t_{k_A}) = \texttt{true}, \quad i = 1, \ldots, n(A).$$

Будем говорить, что протокол является безопасным для субъекта A начиная с некоторого момента времени t_{k_A}, где $k_A \in \{1, \ldots, k_{\max}\}$, если:

1) для всех $k \geqslant k_A$ протокол обладает подтверждённым состоянием $A(t_k)$ субъекта A;
2) выполнено одно из двух утверждений:
   — вероятность π успешной компрометации протокола удовлетворяет неравенству $\pi < \pi_0$;
   — трудоёмкость Q компрометации протокола удовлетворяет неравенству $Q > Q_0$.

Будем называть протокол безопасным начиная с некоторого момента времени t_k, если он является безопасным для каждого участвующего в информационном взаимодействии субъекта $A, B, \ldots$ начиная с момента времени t_{K_A}, t_{K_B}, $\ldots$ соответственно и $t_k \geqslant \max\{t_{K_A}, t_{K_B}, \ldots\}$.

4. Методика оценки безопасности

Теперь мы можем сформулировать методику оценки безопасности созданного ранее или вновь разрабатываемого криптографического протокола.

Область применения методики: криптографические протоколы выработки общего ключа, а также транспортные криптографические протоколы, предназначенные для передачи конфиденциальной информации между субъектами взаимодействия.

Исходными данными для проведения исследования являются:

— класс средств защиты, в которых предполагается использование исследуемого протокола (в случае, если класс средств не определён, анализ должен проводиться для максимального класса средства защиты (см. рекомендации Р 1323565.012-2017));
— модель угроз и модель нарушителя, относительно которых оценивается эффективность защиты, обеспечиваемой исследуемым протоколом (в случае, если модель угроз не определена, должна использоваться модель согласно [1]; если не определена модель нарушителя, то должна использоваться модель, приведённая в рекомендациях Р 1323565.012-2017 для выбранного класса средств защиты информации);
— допустимые значения показателей эффективности защиты π_0 и Q_0, определённые действующими требованиями по безопасности для выбранного класса средств защиты информации (если такие требования существуют);
— спецификация протокола, в соответствии с которой предполагается его практическая реализация;
— условия практической эксплуатации протокола — к таким данным могут относиться сведения о пропускной способности канала, объёме и допустимом времени передачи конфиденциальной информации, сроках смены ключевой информации, допустимом числе ложных попыток аутентификации и т. п.

В результате проведения исследования будут получены:

— перечень свойств безопасности, обеспечиваемых исследуемым протоколом (допускается ситуация, при которой данный перечень свойств может оказаться пустым);
— численные значения показателей эффективности защиты π и Q;
— если заданы допустимые значения показателей эффективности защиты π_0 и Q_0, то заключение о безопасности или небезопасности исследуемого протокола.

Последовательность исследований должна состоять из следующих шагов:

1. Необходимо построить формальную модель протокола, описывающую состояния каждого субъекта взаимодействия и преобразования, применяемые к ячейкам памяти и поступающим из канала связи данным, т. е.

— определить используемые в протоколе криптографические преобразования;
— определить ключевую и криптографически опасную информацию, в частности определить процедуры выработки сессионной и производной ключевой информации;
— определить множество ячеек памяти, образующих состояния участвующих во взаимодействии субъектов; в качестве значений, помещаемых в ячейки памяти, должны выступать
  — исходная ключевая информация;
  — случайные значения, вырабатываемые субъектами в ходе взаимодействия;
  — производная ключевая информация;
  — а также значения, отличные от указанных выше и используемые для выработки производной ключевой информации;

  дополнительно, в состояние субъекта могут включаться значения, определяемые спецификацией протокола, которые, по мнению исследователя, могут влиять на значения определяемых показателей эффективности защиты;
— определить число возможных состояний каждого субъекта (с учётом действующих требований по безопасности, накладывающих ограничения на использование ключевой информации);
— определить области допустимых значений для каждой из случайных величин $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, получаемых субъектом из канала связи или генератора случайных величин (для всех возможных значений индекса k);
— определить отображения, задающие переход субъекта из одного состояния в другое, включая функции изменения значений ячеек памяти и функции, подтверждающие эти значения.

После построения модели протокола должно быть показано, что все ячейки состояний каждого из субъектов взаимодействия должны быть подтверждены явно или косвенно (см. определение 5). При синтезе нового протокола это свойство должно выполняться в обязательном порядке.

Если для разработанного ранее протокола это свойство не выполнено, то появляется возможность построения атаки на протокол, направленной на навязывание неподтверждённого значения. Для поиска таких атак должны быть применены средства автоматизированной верификации протоколов, такие, как Scyther [43], Proverif [44], Avispa [32] или им подобные.

2. Необходимо рассмотреть приведённый в п. 1 перечень свойств безопасности и удалить из него свойства, неприменимые к исследуемому протоколу в связи с выбранной моделью угроз и условиями практического применения протокола. Для оставшихся в перечне свойств должна быть проведена проверка их выполнимости в соответствии

с формальными определениями, сформулированными в п. 2, а также с учётом зависимостей, указанных в табл. 3.

Если для проверки выполнимости какого-либо свойства безопасности должны быть определены значения трудоёмкости Q и вероятности успеха π обращения одной или нескольких однонаправленных функций, то такие значения должны определяться с учётом условий практической эксплуатации протокола, содержащихся в исходных данных для проведения исследования.

3. В обязательном порядке должен быть проведен анализ используемой ключевой информации, включающий в себя рассмотрение следующих вопросов:

— Перед началом выполнения протокола субъектам должна быть доступна исходная ключевая информация, используемая для аутентификации сторон взаимодействия (ключи аутентификации). Если используются симметричные ключи, то они должны быть предварительно распределены с использованием организационно-технических мер защиты. Если используются асимметричные пары ключей, то открытые ключи должны быть подтверждены электронной подписью удостоверяющего (доверенного) центра, а открытые ключи удостоверяющего центра должны быть доставлены субъектам с использованием организационно-технических мер защиты. Отсутствие подтверждённой исходной ключевой информации приводит к нарушению свойства аутентификации субъекта (свойство С 1) и, как следствие, к нарушению большинства из рассмотренных свойств безопасности.

— Необходимо, чтобы ключи аутентификации не использовались непосредственно для шифрования и имитозащиты передаваемой информации. В противном случае возможно как исчерпание ресурса ключа, так и реализация нарушителем атак, использующих конфиденциальную информацию для нарушения свойства аутентификации.

— Основное требование к производной ключевой информации, применяемой для шифрования и имитозащиты передаваемых сообщений, заключается в невозможности её определения нарушителем с трудоёмкостью, меньшей чем тотальное опробование всех возможных значений. Каждый ключ, как правило, представляется в виде двоичного вектора длины m бит, таким образом, нарушителю необходимо опробовать 2^m ключей для компрометации сообщений. Отсюда следует, что необходимо проверить выполнимость следующих условий:
  — множество значений, которые может принимать производный ключ, совпадает с множеством $\mathbb{V}_m$;
  — принимаемые производным ключом значения непредсказуемы, т. е. последовательность нескольких выработанных в различных сессиях протокола производных ключей $K_1, K_2, \ldots$ должна быть статистически неотличима от последовательности случайных равновероятно распределённых на множестве $\mathbb{V}_m$ величин.

— При практическом применении средств защиты информации могут нарушаться правила эксплуатации средств, превышаться заданные ограничения на объём обрабатываемой информации или возникать уязвимости в программном обеспечении, все вместе или по отдельности приводящие к возможности практического определения нарушителем производных ключей или исходной ключевой информации. Это приводит к необходимости встраивания в криптографические протоколы мер, минимизирующих объём скомпрометированной информации. В качестве таких мер могут выступать:

  - использование односторонних функций, не позволяющих вычислять значения ключей аутентификации по значениям производных ключей;
  - использование в каждой сессии протокола уникальных случайных значений для выработки производных ключей (свойства С 12, С 13);
  - использование «древовидных» структур выработки производных ключей, не позволяющих нарушителю по известному производному ключу K_n вычислить значения ключей K_{n-1} и K_{n+1} (свойство С 11).
- Дополнительно в рамках математических исследований должны быть проверены следующие гипотезы:
  - о ничтожной вероятности совпадения различных производных ключей, вырабатываемых в рамках одной сессии протокола;
  - о статистической независимости последовательности производных ключей $K_1, K_2, \ldots$, вырабатываемых в различных сессиях протокола.

4. Необходимо проверить, возможно ли применение известных ранее атак для компрометации исследуемого протокола. Для этого необходимо, во-первых, подготовить базу известных атак на криптографические протоколы из рассматриваемого класса, а во-вторых, провести классификацию известных атак, проведя систематизацию по следующим принципам:

- по методам реализации атаки; к таким методам могут быть отнесены повтор или отражение передаваемых сообщений, использование задержек и перемешивание передаваемых сообщений, изменение формата передаваемых сообщений, использование сообщений из других сессий и т. п.;
- по объектам проведения атаки; в качестве объектов атаки могут выступать передаваемые данные, секретные ключи, случайные значения, вырабатываемые в ходе протокола, и т. п.;
- по свойствам безопасности, поскольку каждая успешно применимая атака приводит к нарушению одного или нескольких свойств безопасности;
- техническим возможностям, необходимым для проведения атаки, например возможностям по перехвату передаваемых данных;
- месту проведения атаки: может ли данная атака проводиться внешним нарушителем или внутренним.

Использование подобной классификации позволяет сузить перечень атак, которые могут быть применены для компрометации исследуемого протокола. Действительно, если протокол содержит явно прописанные в спецификации меры защиты от атак повтором, то такой класс атак может оказаться неприменимым. Аналогично, атака на идентификатор субъекта взаимодействия может оказаться неприменимой, если моделью угроз определено, что данный идентификатор представляет собой общедоступную информацию. Удостоверившись в выполнении определённого ранее перечня свойств безопасности, также можно отсеять часть атак на исследуемый протокол. Для оставшихся атак должна быть показана невозможность их применения либо предложен способ компрометации исследуемого протокола.

5. С использованием описанного в п. 3 метода должны быть определены численные значения показателей эффективности защиты информации — вероятности успешной компрометации протокола π и трудоёмкости успешной компрометации Q. Данные величины должны быть получены для всех способов компрометации исследуемого протокола, предложенных в ходе четвёртого шага исследования. После чего, согласно определению 16, должен быть сделан вывод о безопасности протокола. Отметим, что

полнота проводимого исследования может быть достигнута только в случае выполнения всех перечисленных шагов.

Предложенная методика обладает следующими достоинствами:

— методика позволяет определить конкретные численные значения показателей эффективности защиты; следует добавить, что на настоящий момент времени авторам не известен какой-либо другой поход к определению значений рассматриваемых показателей защиты;
— результаты исследования могут быть использованы при сертификации средств криптографической защиты информации, реализующих исследуемый протокол.

Также можно указать ряд недостатков предложенной методики исследования:

— поскольку при проведении анализа рассматриваются только известные атаки на криптографические алгоритмы и протоколы, то всегда существует вероятность, что найдётся атака, имеющая сложность меньше чем у атак, рассмотренных в ходе исследования; таким образом, полученные значения показателей эффективности должны рассматриваться не как точные значения, а как оценки сверху;
— несогласованность с положениями теории «доказуемой стойкости», принятой в зарубежных изданиях.

5. Анализ протоколов семейства IPSec

В 2021 г. авторами настоящей работы было успешно завершено исследование семейства протоколов, составляющих архитектуру безопасности сети Интернет (Internet Protocol security architecture, IPsec [54, 55]). Данное семейство предназначено для обеспечения защищённого обмена IP-пакетами и включает в себя два основных протокола:

— протокол IKEv2, предназначенный для выработки общей ключевой информации, взаимной аутентификации субъектов взаимодействия и создания/удаления защищённых соединений;
— протокол ESP, обеспечивающий непосредственную защиту передаваемых IP-пакетов в рамках одной сессии защищённого взаимодействия.

Для иллюстрации сказанного в предыдущих пунктах построим формальную модель входящих в IPSec криптографических протоколов.

5.1. Описание протоколов семейства IPSec

В основе протокола выработки общей ключевой информации IKEv2 лежит схема выработки общего ключа «Сигма» [56], реализуемая в группе точек эллиптической кривой $\mathcal{E}$, порождённой точкой P простого порядка q. Взаимодействие между субъектами реализуется путём отправки и получения пары сообщений (по схеме «запрос — ответ»). В качестве канала связи используется протокол UDP.

Для аутентификации сторон взаимодействия используются ключи аутентификации и проверки кода аутентификации субъектов (K_{aA}, K_{cA} для субъекта A и K_{aB},K_{cB} для субъекта B), а также K_u — ключ проверки кода аутентификации удостоверяющего центра. Помимо перечисленных ключей аутентификации, в протоколе IKEv2 дополнительно вырабатывается и применяется производная ключевая информация, приведённая в табл. 4.

В ходе первого сообщения субъект A, инициирующий выполнение протокола, направляет субъекту B следующую информацию: $\xi_1(t_1)$ — уникальный идентификатор соединения $A \to B$, выступающий в качестве заголовка сообщений, отправляемых от субъекта A к субъекту B; $\xi_2(t_1)$ — уникальный двоичный вектор и $\xi_3(t_2)$ — точка эллиптической кривой, используемые для выработки производной ключевой информации;

Таблица 4

Ключевая информация протокола IKEv2

Информация	Назначение	Вычисляется из	Время жизни
$\xi_3(t_1)$, $\xi_3(t_3)$	Случайные значения, вырабатываемые независимо A и B	ДСЧ	Удаляются сразу после использования
Q_{AB}	Общая точка эллиптической кривой $\mathcal{E}$	$\xi_3(t_1)$, $\xi_3(t_3)$	Удаляется сразу после использования
$SKEYSEED$	Общая ключевая информация	$\xi_2(t_1)$, $\xi_2(t_3)$, Q_{AB}	Удаляется после выработки производных ключей
SK_d	Вычисление производных ключей	$SKEYSEED$ и $\xi_2(t_1)$, $\xi_2(t_3)$, $\xi_1(t_1)$, $\xi_1(t_3)$	Удаляется после закрытия защищённого соединения
iSK_A	Имитозащита сообщений $A \to B$	См. выше	См. выше
iSK_B	Имитозащита сообщений $B \to A$	См. выше	См. выше
eSK_A	Шифрование сообщений $A \to B$	См. выше	См. выше
eSK_B	Шифрование сообщений $B \to A$	См. выше	См. выше
aSK_A	Аутентификация субъекта A	См. выше	См. выше
aSK_B	Аутентификация субъекта B	См. выше	См. выше
K_{encr}	Исходная ключевая информация для ESP	См. выше	Время жизни протокола ESP

$\xi_4(t_2)$ — перечень параметров безопасности, используемых для аутентификации субъектов взаимодействия.

При получении сообщения субъект B выбирает приемлемый для себя набор параметров безопасности $\xi_4(t_4)$, вырабатывает $\xi_1(t_3)$ — уникальный идентификатор соединения $B \to A$, $\xi_2(t_3)$ — уникальный двоичный вектор, а также свою точку эллиптической кривой $\xi_3(t_4)$, используемую для выработки общей ключевой информации. Первая пара сообщений, которую принято называть этапом инициализации защищённого соединения, представлена на рис. 15.

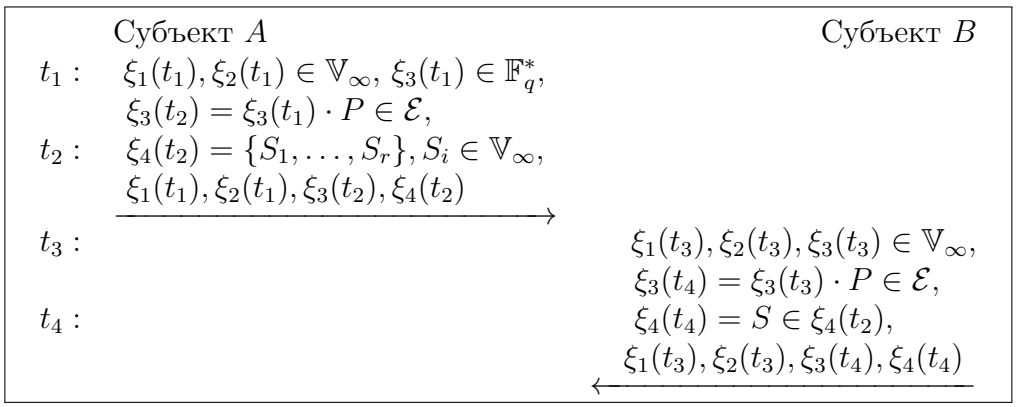

Рис. 15. Схема этапа инициализации протокола IKEv2

После этапа инициализации субъекты приступают к выработке ключевой информации. Вычисляется общая точка эллиптической кривой

$$Q_{AB} = \xi_3(t_1)\xi_3(t_3) \cdot P,$$

далее вычисляется общая ключевая информация $SKEYSEED$=$\texttt{prf}(\xi_2(t_1)||\xi_2(t_3), Q_{AB})$ и производные ключи

$$\begin{gathered} SK_d||iSK_A||iSK_B||eSK_A||eSK_{eB}||aSK_{pA}||aSK_B = \\ = \texttt{prf+}(SKEYSEED, \xi_2(t_1)||\xi_2(t_3)||\xi_1(t_1)||\xi_1(t_3)), \end{gathered} \tag{8}$$

где $\texttt{prf}(K, S) : \mathbb{K} \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ есть функция hmac [57] выработки имитовставки длиной m бит, а prf+ представляет собой итеративное отображение, вырабатывающее псевдослучайную последовательность с использованием функции prf:

$$\texttt{prf+}(K, S) = T_1||T_2||T_3 \ldots,$$
$$\text{где } \; T_1 = \texttt{prf}(K, S||\texttt{0x01}), \;\; T_2 = \texttt{prf}(K, T_1||S||\texttt{0x02}), \;\; T_3 = \texttt{prf}(K, T_2||S||\texttt{0x03}), \;\; \ldots$$

Исходная ключевая информация K_{encr} для протокола ESP определяется равенством

$$K_{\text{encr}} = \texttt{prf+}(SK_d, Q_{AB}||\xi_2(t_1)||\xi_2(t_3)). \tag{9}$$

Отметим, что все случайные значения, используемые для генерации ключевой информации, вырабатываются субъектами A и B независимо друг от друга. Схема выработки ключевой информации представлена на рис. 16.

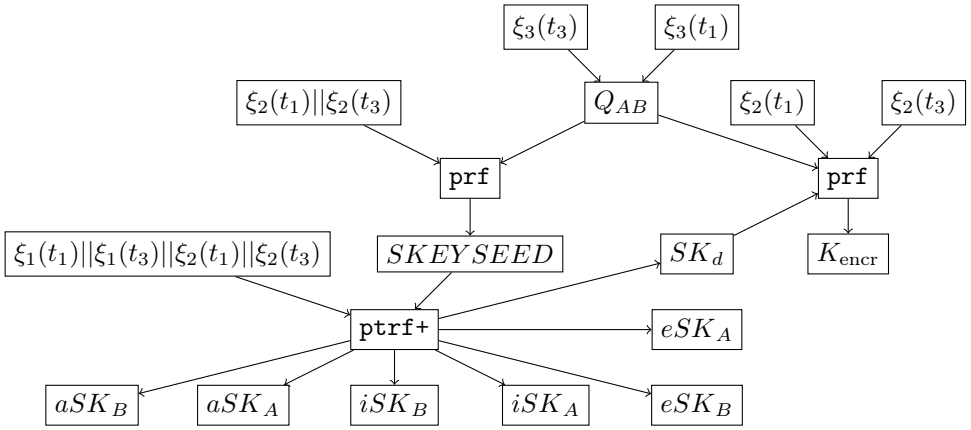

Рис. 16. Иерархия ключевой информации протокола IKEv2

После выработки ключевой информации субъекты переходят к этапу аутентификации. Субъект A направляет субъекту B следующую информацию: $\xi_1(t_1)$ — уникальный идентификатор соединения $A \to B$, выступающий в качестве заголовка сообщений, отправляемых от субъекта A к субъекту B; $\xi_1(t_5)$ — зашифрованное сообщение M_A, содержащее идентификатор ID_A субъекта A; $\texttt{Cert}(K_{aA})$ — сертификат ключа проверки кода аутентификации субъекта A (в состав сертификата должны входить значение ключа K_{aA} и электронная подпись под значением ключа, выработанная удостоверяющим центром) и уникальную метку $AUTH_A$, используемую для аутентификации

субъекта A; $\xi_2(t_5)$ — код аутентификации отправляемого сообщения. Метка $AUTH_A$ определяется равенствами

$$\begin{aligned} BLOB_A &= \underbrace{\xi_1(t_1)||\xi_2(t_1)||\xi_3(t_2)||\xi_4(t_2)}_{\text{см. рис.15}}||\xi_2(t_3)||\mathtt{prf}(aSK_A, ID_A), \\ AUTH_A &= \mathtt{sign}(K_{aA}, BLOB_A), \end{aligned} \tag{10}$$

где $\mathtt{sign}$ — функция выработки электронной подписи, а подписываемое сообщение $BLOB_A$ зависит не только от отправленного ранее субъектом A сообщения $\xi_1(t_1)$, $\xi_2(t_1)$, $\xi_3(t_2)$, $\xi_4(t_2)$ и выработанного субъектом B случайного значения $\xi_2(t_3)$, но и от производного ключа aSK_A.

Далее субъект B выполняет следующую последовательность шагов:

— расшифровывает полученное сообщение и проверяет его корректность, т. е. выполнение условия $\mathtt{conf}(iSK_A, \mathtt{dec}(eSK_A, \xi_1(t_5)), \xi_2(t_5)) =^? \mathtt{true}$; если условие не выполнено, протокол прерывается с уведомлением о неудаче;

— выделяет из расшифрованного сообщения K_{cA} — ключ проверки кода аутентификации субъекта A — и подтверждает его с помощью ключа доверенного центра K_u, т. е. проверяет выполнение условия $\mathtt{verify}(K_u, \mathtt{Cert}(K_{cA})) =^? \mathtt{true}$; если условие не выполнено, протокол прерывается с уведомлением о неудаче;

— выделяет из расшифрованного сообщения идентификатор ID_A, формирует в соответствии с (10) сообщение $BLOB_A$ и проверяет его корректность, т. е. выполнение условия $\mathtt{verify}(K_{cA}, BLOB_A, AUTH_A) =^? \mathtt{true}$; если условие не выполнено, протокол прерывается с уведомлением о неудаче;

— в случае успешного завершения всех проверок аутентифицирует субъекта A.

После этого для собственной аутентификации субъект B направляет аналогичное сообщение субъекту A. Схема этапа аутентификации представлена на рис. 17.

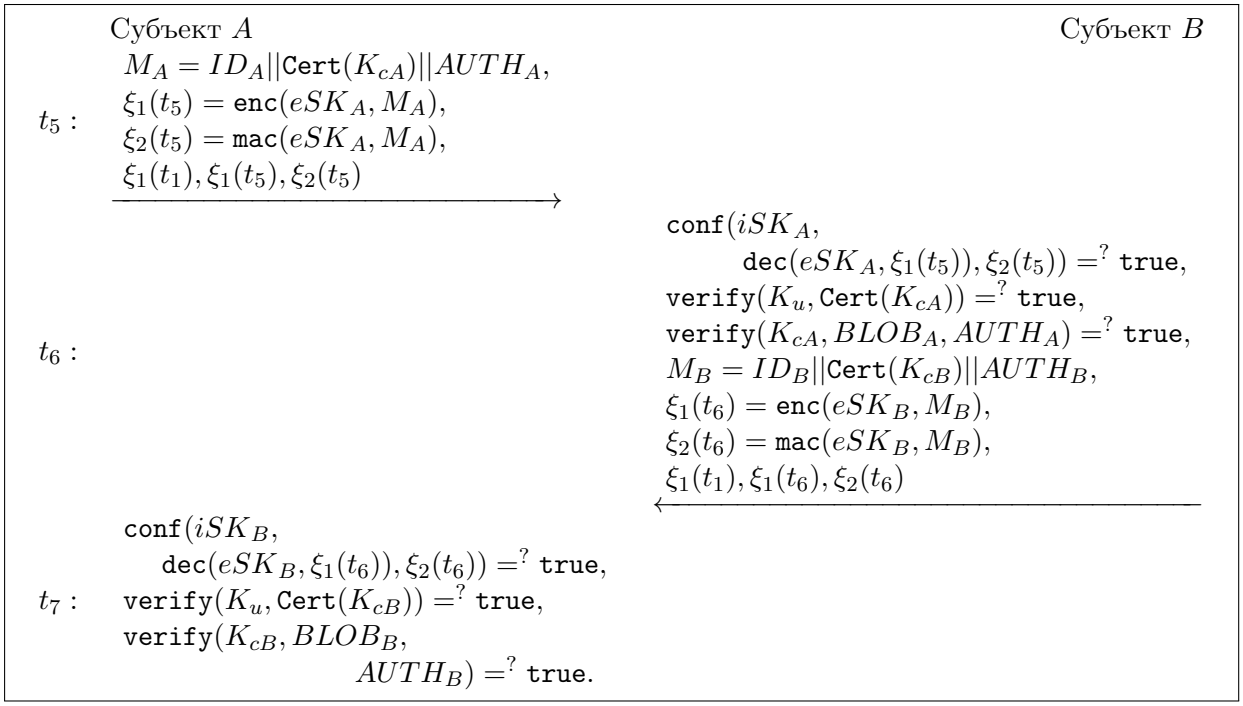

Рис. 17. Схема этапа аутентификации протокола IKEv2

После того как субъект A завершает аутентификацию субъекта B, начинается процесс обмена зашифрованными IP-пакетами по каждому из созданных однонаправленных соединений — в первом соединении сообщения направляются от субъекта A к субъекту B, во втором соединении — в обратную сторону. Механизм шифрования и имитозащиты передаваемых пакетов определяется протоколом ESP.

Согласно Р 1323565.1.030-2021, протокол ESP использует выработанную в ходе выполнения протокола IKEv2 ключевую информацию K_{encr} (см. табл. 4) для генерации производных ключей шифрования передаваемых IP-пакетов.

Пусть $K_{\text{encr}} = K_1||K_s$, где двоичная длина K_s зависит от длины блока используемого блочного шифра. Производные ключи вырабатываются согласно равенству

$$K_{\text{msg}} = \texttt{tree}(K_1, i_1, i_2, i_3) = \texttt{kdf}(\texttt{kdf}(\texttt{kdf}(K_1, l_1, \texttt{0x00}||i_1), l_2, i_2),$$

где функция $\texttt{kdf}(K, L, S) : \mathbb{K} \times \mathbb{V}_\infty^* \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ построена с использованием функции `hmac` (см. Р.50.1.113-2016):

$$\texttt{kdf}(K, L, S) = \texttt{hmac}(K, \texttt{0x01}||L||\texttt{0x00}||S||\texttt{0x00}||\texttt{0x01});$$

l_1, l_2, l_3 — фиксированные константы из $\mathbb{V}_\infty$; $i_1 \in \mathbb{Z}_{2^8}$; $i_2, i_3 \in \mathbb{Z}_{2^{16}}$.

На каждом производном ключе K_{msg} может быть зашифровано несколько пакетов, которые нумеруются с помощью счётчика N_p, $N_p \in \{0, \ldots, 2^{24}\}$; в протоколе ESP с помощью счётчика N_s нумеруются также все передаваемые в рамках одного соединения пакеты. Счетчик N_s используется для защиты от атак навязывания повторных пакетов, а счётчик N_p — для контроля объёма информации, зашифрованной на одном ключе K_{msg}. Шифрование сообщения $M \in \mathbb{V}_\infty$ (IP-пакета) осуществляется согласно схеме рис. 18.

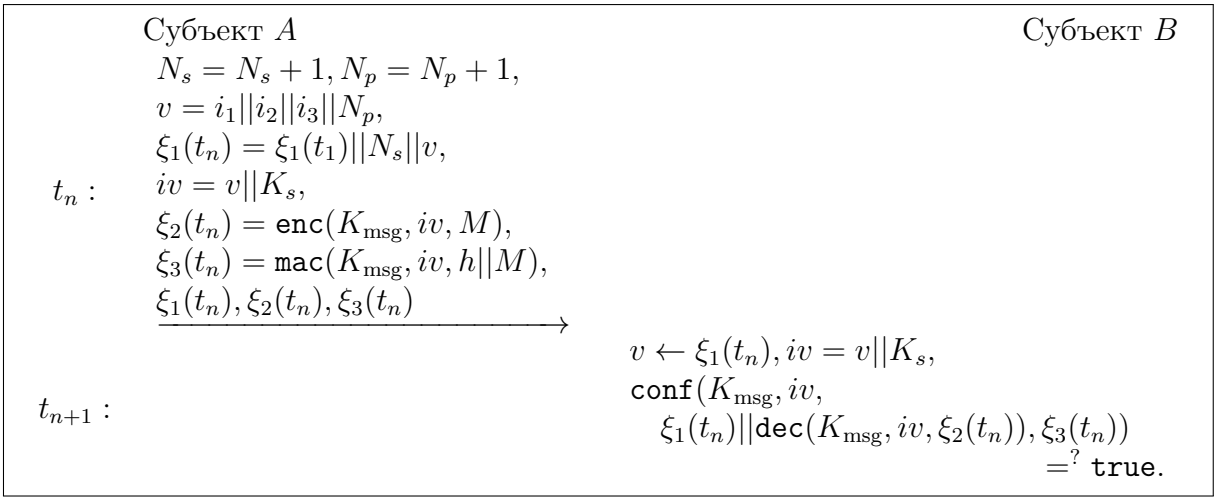

Рис. 18. Схема протокола ESP

5.2. Построение формальной модели

Построим модель состояний субъекта A, инициализирующего выполнение протокола IKEv2. Из рис. 17 видно, что количество возможных состояний субъекта A в процессе выполнения протокола равно 7, последующие состояния будем относить к протоколу ESP. В табл. 5 перечислено множество ячеек памяти, образующих $A(t_0)$ — состояние

субъекта A в момент времени t_0. Ячейки $\alpha_1, \ldots, \alpha_{22}$ относятся к протоколу IKEv2, ячейки $\alpha_{23}, \ldots, \alpha_{25}$ — к протоколу ESP.

Таблица 5

Множество состояний субъекта A

Ячейка	Значение	Ячейка	Значение
$\alpha_1(ID_A, \texttt{true})$	Идентификатор субъекта A	$\alpha_2(\varnothing, \texttt{false})$	Идентификатор субъекта B
$\alpha_3(K_{aA}, \texttt{true})$	Ключ аутентификации субъекта A	$\alpha_4(\varnothing, \texttt{false})$	Ключ проверки кода аутентификации субъекта B
$\alpha_5(K_u, \texttt{true})$	Ключ проверки кода аутентификации удостоверяющего центра	$\alpha_6(\varnothing, \texttt{false})$, $\alpha_7(\varnothing, \texttt{false})$	Идентификатор соединения $\xi_1(t_1)$ и случайное значение $\xi_2(t_1)$
$\alpha_8(\varnothing, \texttt{false})$	Случайное значение $\xi_3(t_1)$	$\alpha_9(\varnothing, \texttt{false})$, $\alpha_{10}(\varnothing, \texttt{false})$	Идентификатор соединения $\xi_1(t_3)$ и случайное значение $\xi_2(t_3)$ субъекта B
$\alpha_{11}(\varnothing, \texttt{false})$	Точка кривой $\xi_3(t_4)$, выработанная субъектом B	$\alpha_{12}(\varnothing, \texttt{false})$	Множество параметров безопасности $\xi_4(t_4)$
$\alpha_{13}(\varnothing, \texttt{false})$	Общая точка Q_{AB}	$\alpha_{14}(\varnothing, \texttt{false})$	Общая ключевая информация $SKEYSEED$
$\alpha_{15}(\varnothing, \texttt{false})$	Ключ SK_d	$\alpha_{16}(\varnothing, \texttt{false})$	Ключ iSK_A
$\alpha_{17}(\varnothing, \texttt{false})$	Ключ iSK_B	$\alpha_{18}(\varnothing, \texttt{false})$	Ключ eSK_A
$\alpha_{19}(\varnothing, \texttt{false})$	Ключ eSK_B	$\alpha_{20}(\varnothing, \texttt{false})$	Ключ aSK_A
$\alpha_{21}(\varnothing, \texttt{false})$	Ключ aSK_B	$\alpha_{22}(\varnothing, \texttt{false})$	Ключ K_{encr}
$\alpha_{23}(\varnothing, \texttt{false})$	Ключ K_{msg}	$\alpha_{24}(\varnothing, \texttt{false})$	Синхропосылка
$\alpha_{25}(\varnothing, \texttt{false})$	Принимаемое сообщение		

Последовательность последующих состояний субъекта A описывается следующими нетривиальными функциями:

$$
\begin{aligned}
A(t_1)\colon\quad & \alpha_6 \leftarrow (\xi_1(t_1), \texttt{true}),\ \alpha_7 \leftarrow (\xi_2(t_1), \texttt{true}),\ \alpha_8 \leftarrow (\xi_3(t_1), \texttt{true});\\
A(t_5)\colon\quad & \alpha_9 \leftarrow (\xi_1(t_3), \texttt{false}),\ \alpha_{10} \leftarrow (\xi_2(t_3), \texttt{false}), \alpha_{11} \leftarrow (\xi_3(t_4), \texttt{true}),\\
& \alpha_{12} \leftarrow (\xi_4(t_4), \texttt{false}), \alpha_2 \leftarrow (ID_B, \sigma),\\
& \alpha_4 \leftarrow (K_{cB},\ \texttt{verify}(a_5, \texttt{Cert}(K_{cB}) \leftarrow \xi_1(t_6))),\\
& \alpha_{13} \leftarrow (Q_{AB}, \sigma),\ \alpha_{14} \leftarrow (SKEYSEED, \sigma),\ \alpha_{15} \leftarrow (SK_d, \texttt{false});\\
A(t_7)\colon\quad & \alpha_{16} \leftarrow (iSK_A, \texttt{false}),\ \alpha_{17} \leftarrow (iSK_B, \texttt{false}),\\
& \alpha_{18} \leftarrow (eSK_A, \texttt{false}),\ \alpha_{19} \leftarrow (eSK_B, \texttt{false}),\ \alpha_{20} \leftarrow (aSK_A, \texttt{false}),\\
& \alpha_{21} \leftarrow (aSK_B, \sigma),\ \alpha_{22} = (K_{\text{encr}}, \texttt{false}),\ \alpha_{23} = (K_{\text{msg}}, \texttt{false}),\\
& \{\sigma_9, \sigma_{10}, \sigma_{12}\} \leftarrow \sigma,
\end{aligned}
$$

где $\sigma = \texttt{verify}(a_4, \underbrace{a_9||a_{10}||a_{11}||a_{12}||a_7||\texttt{prf}(aSK_B, ID_B \leftarrow \xi_1(t_6))}_{BLOB_B}, AUTH_B \leftarrow \xi_1(t_6))$.

Легко видеть, что по завершении протокола IKEv2 значения ячеек $\alpha_{15}, \ldots, \alpha_{20}$, α_{22}, α_{23} не являются подтверждёнными. С другой стороны, поскольку они однозначно выражаются через подтверждённые значения (см. (8) и (9)), будем говорить, что значения, содержащиеся в ячейках $\alpha_{15}, \ldots, \alpha_{20}, \alpha_{22}, \alpha_{23}$, подтверждены косвенно. Это же можно сказать и о производном ключе K_{msg}, используемом в протоколе ESP.

Изменение состояний субъекта A, получающего сообщения от субъекта B в ходе выполнения протокола ESP, для всех $k \geqslant 8$ описывается следующим образом:

$$
\begin{aligned}
&\alpha_{24} = v \leftarrow \xi_1(t_k)\ ||\ K_s \leftarrow a_{22},\\
&\alpha_{25} = \texttt{dec}(\alpha_{23}, \alpha_{24}, \xi_2(t_k)),\\
&\sigma_{24} = \sigma_{25} = \texttt{conf}(\alpha_{23}, \alpha_{24}, \xi_1(t_n)||\alpha_{25}, \xi_3(t_k)).
\end{aligned}
$$

Построенная модель позволяет говорить, что для протокола IKEv2 выполнены следующие свойства безопасности: С 1, С 2, С 9, С 10, С 15, С 16, С 18 (в части идентификатора инициатора протокола), С 3, С 27. Исследуя процедуры выработки производной ключевой информации, можно показать, что для протокола IKEv2 выполнены свойства С 11, С 12, С 13, С 14. Протокол ESP наследует указанные свойства, дополнительно обеспечивает свойства С 4, С 17 и содержит механизмы, необходимые для выполнения свойств С 20 и С 21.

Трудоёмкость компрометации протокола IKEv2 существенно зависит от используемых криптографических преобразований и следует из трудоёмкости решения задачи дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}$ простого порядка q (см. рис. 15), задачи определения секретного ключа eSK_A или eSK_B по известным нарушителю значениям $\xi_1(t_5), \xi_2(t_5)$ или $\xi_1(t_6), \xi_2(t_6)$, см. рис. 17, задачи подделки кодов аутентификации $\xi_2(t_5)$ или $\xi_2(t_6)$ при неизвестном значении аутентифицируемого сообщения M_A или M_B (см. рис. 17), задачи подделки значения электронной подписи под сообщениями $BLOB_A$ или $BLOB_B$ (см. равенства (10)).

Для компрометации протокола ESP нарушителю достаточно уметь решать задачи определения ключа шифрования K_{msg} и навязывания кода аутентификации сообщения $\xi_3(t_n)$, см. рис. 18.

Авторы выражают благодарность В. А. Смыслову, С. В. Матвееву и В. Н. Цыпышеву за содержательные замечания и помощь при проведении анализа протоколов семейства IPSec.

ЛИТЕРАТУРА

1. ГОСТ Р ИСО/МЭК 27033-1:2011. Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Ч. 1. Обзор и концепции. М.: Стандартинформ, 2012. 73 с.
2. Р 1323565.1.012-2017. Информационная технология. Криптографическая защита информации. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации. М.: Стандартинформ, 2017. 28 с.
3. *Dolev D. and Yao A.* On the security of public key protocols // IEEE Trans. Inform. Theory. 1983. V. 29. No. 2. P. 198–208.
4. *Basin D. and Cremers C.* Modeling and analyzing security in the presence of compromising adversaries // LNCS. 2010. V. 6345. P. 340–356.
5. *Lowe G.* Breaking and fixing the Needham — Schroeder Public-Key Protocol using FDR // LNCS. 1996. V. 1055. P. 1–20.
6. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. `https://docs.cntd.ru/document/1200058320`. 2008.
7. *Bellare M. and Rogaway P.* Entity authentication and key distribution // LNCS. 1993. V. 773. P. 232–249.
8. *Bellare M., Pointcheval D., and Rogaway P.* Authenticated key exchange secure against dictionary attacks // LNCS. 2000. V. 1807. P. 139–155.
9. *Bellare M. and Rogaway P.* Provably secure session key distribution — the three party case // 27th ACM Symp. Theory Computing. ACM Press, 1995. P. 57–66.
10. *Blake-Wilson S., Johnson D., and Menezes A.* Key agreement protocols and their security analysis // LNCS. 1997. V. 1355. P. 30–45.
11. *Blake-Wilson S. and Menezes A.* Entity authentication and authenticated key transport protocols employing asymmetric techniques // LNCS. 1998. V. 1361. P. 137–158.

12. *Canetti R. and Krawczyk H.* Analysis of key-exchange protocols and their use for building secure channels // LNCS. 2001. V. 2045. P. 453–474.
13. *LaMacchia B., Lauter K., and Mityagin A.* Stronger security of authenticated key exchange // LNCS. 2007. V. 4784. P. 1–16.
14. *Krawczyk H.* HMQV: A high-performance secure Diffie — Hellman protocol // LNCS. 2005. V. 3621. P. 546–566.
15. *Menezes A. and Ustaoglu B.* On the importance of public-key validation in the MQV and HMQV key agreement protocols // LNCS. 2006. V. 4329. P. 133–147.
16. *Rabin M.* Digitized Signatures and Public Key Functions as Intractable as Factorization. Technical Report: MIT/LCS/TR-212. MIT Laboratory for Computer Science, Cambridge, 1979.
17. *Goldwasser S. and Micali S.* Probabilistic encryption // J. Computer System Sci. 1984. V. 28. P. 270–299.
18. *Mao W.* Modern Cryptography: Theory and Practice. Prentice Hall, New Jersey, 2003. 707 p.
19. *Boyd C., Mathuria A., and Stebila D.* Protocols for Authentication and Key Establishment. Second Ed. Berlin; Heidelberg: Springer Verlag, 2020. 521 p.
20. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2002. 480 с.
21. *Бабаш А. В., Шанкин Г. П.* Криптография. М.: Солон-Пресс, 2007. 512 с.
22. *Алексеев Е. К., Ахметзянова Л. Р., Ошкин И. Б., Смышляев С. В.* Обзор уязвимостей некоторых протоколов выработки общего ключа с аутентификацией на основе пароля и принципы построения протокола SESPAKE // Математические вопросы криптографии. 2016. Т. 7. № 4. С. 7–28.
23. *Ahmetzyanova L. R., Alekseev E. K., Sedov G. K., et al.* Practical significance of security bounds for standardized internally re-keyed block cipher modes // Математические вопросы криптографии. 2019. Т. 10. № 2. С. 31–46.
24. Р 1323565.1.030-2020. Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3). М.: Стандартинформ, 2020. 73 с.
25. Р 1323565.1.028-2018. Информационная технология. Криптографическая защита информации. Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств. М.: Стандартинформ, 2019. 66 с.
26. *Нестеренко А. Ю.* Об одном подходе к построению защищенных соединений // Математические вопросы криптографии. 2013. Т. 4. №. 2. С. 101–111.
27. *Нестеренко А. Ю., Лебедев П. А., Семенов А. М.* Краткий анализ криптографических механизмов защищенного взаимодействия контрольных и измерительных устройств. Технический комитет по стандартизации «Криптографическая защита информации». «Криптографические исследования». 2019. Сер. б/н. `https://tc26.ru/standarts/kriptograficheskie-issledovaniya/`.
28. *Semenov A. M.* Analysis of Russian key-agreement protocols using automated verification tools // Математические вопросы криптографии. 2017. Т. 8. № 2. С. 131–142.
29. Р 1323565.1.035–2021. Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе защиты информации ESP. М.: Стандартинформ, 2021. 52 с.
30. *Черемушкин А. В.* Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. Приложение. 2009. № 2. С. 115–150.
31. IETF. RFC 3552. Guidelines for Writing RFC Text on Security Considerations. 2003. `https://tools.ietf.org/html/rfc3552`.

32. The AVISPA Project. Properties (Goals). 2021. http://www.avispa-project.org/delivs/6.1/d6-1/node3.html.
33. ГОСТ Р 53113.1-2008 Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 1. Общие положения. М.: Стандартинформ, 2008. 12 с.
34. ГОСТ Р 53113.2-2009 Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 2. Рекомендации по организации защиты информации, информационных технологий и автоматизированных систем от атак с использованием скрытых каналов. М.: Стандартинформ, 2009. 12 с.
35. *Видякин В. В.* О связи скрытых информационных каналов и субпротоколов // Обозрение прикл. и промышл. матем. 2006. Т. 13. Вып. 1. С. 87–88.
36. *Князев А. В., Ронжин А. Ф.* Инструментальный анализ мутных протоколов // Обозрение прикл. и промышл. матем. 2007. Т. 14. Вып. 4. С. 577–646.
37. *Матвеев С. В.* Некоторые подходы к оценке пропускной способности скрытых каналов в IP-сетях // Системы высокой доступности. 2012. Т. 8. Вып. 2. С. 68–71.
38. *Blake-Wilson S. and Menezes A.* Unknown key-share attacks on the Station-to-Station (STS) protocol // LNCS. 1999. V. 1560. P. 154–170.
39. *Diffie W., van Oorschot P., and Wiener M.* Authentication and authenticated key exchanges // Des. Codes Crypt. 1992. V. 2. P. 107–125.
40. IETF. RFC 8654. Extended Message Support for BGP. 2019. https://tools.ietf.org/html/rfc8654.
41. IETF. RFC 3748. Extensible Authentication Protocol (EAP). 2004. https://tools.ietf.org/html/rfc3748.
42. IETF. RFC 7029. Extensible Authentication Protocol (EAP) Mutual Cryptographic Binding. 2013. https://tools.ietf.org/html/rfc7029.
43. *Cremers C.* Scyther — Semantics and Verification of Security Protocols. Ph.D. Thesis. Eindhoven Univ. Technology, 2006. 205 p.
44. Proverif: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2020. 150 p. http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf.
45. ГОСТ Р 58833-2020. Защита информации. Идентификация и аутентификация. Общие положения. М.: Стандартинформ, 2020. 28 с.
46. Словарь криптографических терминов / под ред. Б. А. Погорелова и В. Н. Сачкова. М.: МЦМНО. 2006. 94 с.
47. ГОСТ Р ИСО/МЭК 9594-8-98. Информационная технология. Взаимосвязь открытых систем. Справочник. Ч. 8. Основы аутентификации. М.: Стандартинформ, 2001. 29 с.
48. *Fletcher J. G.* An arithmetic checksum for serial transmissions // IEEE Trans. Communications. 1982. V. 30. No. 1. P. 247–252.
49. *Peterson W. W. and Brown D. T.* Cyclic codes for error detection // Proc. IRE. 1961. V. 49. No. 1. P. 228–235. doi:10.1109/JRPROC.1961.287814.
50. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. М.: Стандартинформ, 2012. 25 с.
51. *Alashwali E. and Rasmussen K.* What's in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS. Cryptology ePrint Archive. 2019. Report 2019/1083. https://eprint.iacr.org/2019/1083.
52. *Качалин И. Ф., Кузьмин А. С., Суслов Е. А. и др.* Об основных концепциях криптографической стойкости // Тезисы XII Всерос. школы-коллоквиума по стохастическим ме-

тодам и VI Всерос. симпозиума по прикладной и промышленной математике. Сочи-Дагомыс, 1–7 октября 2005 г. С. 982–983.

53. *Лось А. Б., Нестеренко А. Ю., Рожков М. И.* Криптографические методы защиты информации. М.: Изд-во Юрайт, 2016. 473 с.
54. IETF. RFC 4303. IP Encapsulating Security Payload (ESP). 2005. `https://datatracker.ietf.org/doc/html/rfc4303`.
55. IETF. RFC 7296. Internet Key Exchange Protocol Version 2 (IKEv2). 2014. `https://datatracker.ietf.org/doc/html/rfc7296`.
56. *Krawczyk H.* SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie — Hellman and its use in the IKE protocols // LNCS. 2003. V. 2729. P. 400–425.
57. Р 50.1.113-2016. Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. М.: Стандартинформ, 2016. 28 с.

REFERENCES

1. GOST R ISO/MEK 27033-1:2011. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Bezopasnost' setey. Ch. 1. Obzor i kontseptsii. [Information Technology. Security Techniques. Network Security. P. 1. Overview and Concepts.] Moscow, Standartinform, 2012. 73 p. (in Russian)
2. R 1323565.1.012-2017. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Printsipy razrabotki i modernizatsii shifroval'nykh (kriptograficheskikh) sredstv zashchity informatsii [Information Technology. Cryptographic Data Security. Principles of Creation and Modernization for Cryptographic Modules]. Moscow, Standartinform, 2017. 28 p. (in Russian)
3. *Dolev D. and Yao A.* On the security of public key protocols. IEEE Trans. Inform. Theory, 1983, vol. 29, no. 2, pp. 198–208.
4. *Basin D. and Cremers C.* Modeling and analyzing security in the presence of compromising adversaries. LNCS, 2010, vol. 6345, pp. 340–356.
5. *Lowe G.* Breaking and fixing the Needham — Schroeder Public-Key Protocol using FDR. LNCS, 1996, vol. 1055, pp. 1–20.
6. GOST R 50922-2006. Zashchita informatsii. Osnovnye terminy i opredeleniya [Data Protection. Basic Terms and Definitions]. `https://docs.cntd.ru/document/1200058320`. 2008. (in Russian)
7. *Bellare M. and Rogaway P.* Entity authentication and key distribution. LNCS, 1993, vol. 773, pp. 232–249.
8. *Bellare M., Pointcheval D., and Rogaway P.* Authenticated key exchange secure against dictionary attacks. LNCS, 2000, vol. 1807, pp. 139–155.
9. *Bellare M. and Rogaway P.* Provably secure session key distribution — the three party case. 27th ACM Symp. Theory Computing, ACM Press, 1995, pp. 57–66.
10. *Blake-Wilson S., Johnson D., and Menezes A.* Key agreement protocols and their security analysis. LNCS, 1997, vol. 1355, pp. 30–45.
11. *Blake-Wilson S. and Menezes A.* Entity authentication and authenticated key transport protocols employing asymmetric techniques. LNCS, 1998, vol. 1361, pp. 137–158.
12. *Canetti R. and Krawczyk H.* Analysis of key-exchange protocols and their use for building secure channels. LNCS, 2001, vol. 2045, pp. 453–474.
13. *LaMacchia B., Lauter K., and Mityagin A.* Stronger security of authenticated key exchange. LNCS, 2007, vol. 4784, pp. 1–16.

14. *Krawczyk H.* HMQV: A high-performance secure Diffie — Hellman protocol. LNCS, 2005, vol. 3621, pp. 546–566.
15. *Menezes A. and Ustaoglu B.* On the importance of public-key validation in the MQV and HMQV key agreement protocols. LNCS, 2006, vol. 4329, pp. 133–147.
16. *Rabin M.* Digitized Signatures and Public Key Functions as Intractable as Factorization. Technical Report: MIT/LCS/TR-212, MIT Laboratory for Computer Science, Cambridge, 1979.
17. *Goldwasser S. and Micali S.* Probabilistic encryption. J. Computer System Sci., 1984, vol. 28, pp. 270–299.
18. *Mao W.* Modern Cryptography: Theory and Practice. Prentice Hall, New Jersey, 2003. 707 p.
19. *Boyd C., Mathuria A., and Stebila D.* Protocols for Authentication and Key Establishment. Second Ed. Berlin; Heidelberg, Springer Verlag, 2020. 521 p.
20. *Alferov A. P., Zubov A. Yu., Kuz'min A. S., and Cheremushkin A. V.* Osnovy kriptografii [Fundamentals of Cryptography]. Moscow, Gelios ARV, 2002. 480 p. (in Russian)
21. *Babash A. V. and Shankin G. P.* Kriptografiya [Cryptography]. Moscow, Solon-Press, 2007. 512 p. (in Russian)
22. *Alekseev E. K., Akhmetzyanova L. R., Oshkin I. B., and Smyshlyaev S. V.* Obzor uyazvimostey nekotorykh protokolov vyrabotki obshchego klyucha s autentifikatsiey na osnove parolya i printsipy postroeniya protokola SESPAKE [A review of the password authenticated key exchange protocols vulnerabilities and principles of the SESPAKE protocol construction]. Matematicheskie Voprosy Kriptografii, 2016, vol. 7, no. 4, pp. 7–28. (in Russian)
23. *Ahmetzyanova L. R., Alekseev E. K., Sedov G. K., et al.* Practical significance of security bounds for standardized internally re-keyed block cipher modes. Matematicheskie Voprosy Kriptografii, 2019, vol. 10, no. 2, pp. 31–46.
24. R 1323565.1.030-2020. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Ispol'zovanie kriptograficheskikh algoritmov v protokole bezopasnosti transportnogo urovnya (TLS 1.3) [Information Technology. Cryptographic Data Security. The Use of the Russian Cryptographic Algorithms in the Transport Layer Security Protocol (TLS 1.3)]. Moscow, Standartinform, 2020. 73 p. (in Russian)
25. R 1323565.1.028-2018. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Kriptograficheskie mekhanizmy zashchishchennogo vzaimodeystviya kontrol'nykh i izmeritel'nykh ustroystv [Information Technology. Cryptographic Data Security. Cryptographic Mechanisms of Secure Interactions of Control and Measuring Devices]. Moscow, Standartinform, 2019. 66 p. (in Russian)
26. *Nesterenko A. Yu.* Ob odnom podkhode k postroeniyu zashchishchennykh soedineniy [On an approach to the construction of secure connections]. Matematicheskie Voprosy Kriptografii, 2013, vol. 4, no. 2, pp. 101–111. (in Russian)
27. *Nesterenko A. Yu., Lebedev P. A., and Semenov A. M.* Kratkiy analiz kriptograficheskikh mekhanizmov zashchishchennogo vzaimodeystviya kontrol'nykh i izmeritel'nykh ustroystv [Brief Analysis of Cryptographic Mechanisms of Secure Interaction of Control and Measuring Devices]. Technical Committee on Standardization "Cryptographic Protection of Information". Ser. n/n "Cryptographic research", 2019. `https://tc26.ru/standarts/kriptograficheskie-issledovaniya/`. (in Russian)
28. *Semenov A. M.* Analysis of Russian key-agreement protocols using automated verification tools. Matematicheskie Voprosy Kriptografii, 2017, vol. 8, no. 2, pp. 131–142.
29. R 1323565.1.035–2021. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Ispol'zovanie rossiyskikh kriptograficheskikh algoritmov v protokole zashchity informatsii ESP. [Information Technology. Cryptographic Protection of Information. Using

Russian Cryptographic Algorithms in the ESP Information Protection Protocol]. Moscow, Standartinform, 2021. 52 p. (in Russian)

30. *Cheremushkin A. V.* Kriptograficheskie protokoly: osnovnye svoystva i uyazvimosti [Cryptographic protocols: main properties and vulnerabilities]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2009, no. 2, pp. 115–150. (in Russian)
31. IETF. RFC 3552. Guidelines for Writing RFC Text on Security Considerations, 2003. `https://tools.ietf.org/html/rfc3552`.
32. The AVISPA Project. Properties (Goals), 2021. `http://www.avispa-project.org/delivs/6.1/d6-1/node3.html`.
33. GOST R 53113.1-2008 Informatsionnaya tekhnologiya. Zashchita informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot ugroz informatsionnoy bezopasnosti, realizuemykh s ispol'zovaniem skrytykh kanalov. Ch. 1. Obshchie polozheniya [Information Technology. Protection of Information Technology and Automated Systems against Security Threats Posed by Use of Covert Channels. P. 1. General Principles]. Moscow, Standartinform, 2008. 12 p. (in Russian)
34. GOST R 53113.2-2009 Informatsionnaya tekhnologiya. Zashchita informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot ugroz informatsionnoy bezopasnosti, realizuemykh s ispol'zovaniem skrytykh kanalov. Ch. 2. Rekomendatsii po organizatsii zashchity informatsii, informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot atak s ispol'zovaniem skrytykh kanalov [Information Technology. Protection of Information Technology and Automated Systems against Security Threats Posed by Use of Covert Channels. P. 2. Recommendations on Protecting Information, Information Technology and Automated Systems against Covert Channel Attacks]. Moscow, Standartinform, 2009. 12 p. (in Russian)
35. *Vidyakin V. V.* O svyazi skrytykh informatsionnykh kanalov i subprotokolov [On the connection of hidden information channels and subprotocols]. Obozrenie Prikl. i Promyshl. Matem., 2006, vol. 13, iss. 1, pp. 87–88. (in Russian)
36. *Knyazev A. V. and Ronzhin A. F.* Instrumental'nyy analiz mutnykh protokolov [Instrumental analysis of turbid protocols]. Obozrenie Prikl. i Promyshl. Matem., 2007, vol. 14, iss. 4, pp. 577–646. (in Russian)
37. *Matveev S. V.* Nekotorye podkhody k otsenke propusknoy sposobnosti skrytykh kanalov v IP-setyakh [Some approaches to estimating the bandwidth of hidden channels in IP-networks]. Sistemy Vysokoy Dostupnosti, 2012, vol. 8, iss. 2, pp. 68–71. (in Russian)
38. *Blake-Wilson S. and Menezes A.* Unknown key-share attacks on the Station-to-Station (STS) protocol. LNCS, 1999, vol. 1560, pp. 154–170.
39. *Diffie W., van Oorschot P., and Wiener M.* Authentication and authenticated key exchanges. Des. Codes Crypt., 1992, vol. 2, pp. 107–125.
40. IETF. RFC 8654. Extended Message Support for BGP. 2019. `https://tools.ietf.org/html/rfc8654`.
41. IETF. RFC 3748. Extensible Authentication Protocol (EAP). 2004. `https://tools.ietf.org/html/rfc3748`.
42. IETF. RFC 7029. Extensible Authentication Protocol (EAP) Mutual Cryptographic Binding. 2013. `https://tools.ietf.org/html/rfc7029`.
43. *Cremers C.* Scyther — Semantics and Verification of Security Protocols. Ph.D. Thesis, Eindhoven Univ. Technology, 2006. 205 p.
44. Proverif: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2020. 150 p. `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf`.

45. GOST R 58833-2020. Zashchita informatsii. Identifikatsiya i autentifikatsiya. Obshchie polozheniya [Information Protection. Identification and Authentication. General]. Moscow, Standartinform, 2020. 28 p. (in Russian)
46. Slovar' kriptograficheskikh terminov [Dictionary of Cryptographic Terms]. B. A. Pogorelov and V. N. Sachkov (eds.). Moscow, MCCME publ., 2006. 94 p. (in Russian)
47. GOST R ISO/MEK 9594-8-98. Informatsionnaya tekhnologiya. Vzaimosvyaz' otkrytykh sistem. Spravochnik. Ch. 8. Osnovy autentifikatsii [Information Technology. Open Systems Interconnection. The Directory. P. 8. Authentication Framework]. Moscow, Standartinform, 2001. 29 p. (in Russian)
48. *Fletcher J. G.* An arithmetic checksum for serial transmissions. IEEE Trans. Communications, 1982, vol. 30, no. 1, pp. 247–252.
49. *Peterson W. W. and Brown D. T.* Cyclic codes for error detection. Proc. IRE, 1961, vol. 49, no. 1, pp. 228–235. doi:10.1109/JRPROC.1961.287814.
50. GOST R 34.11-2012. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Funktsiya kheshirovaniya [Information Technology. Cryptographic Data Security. Hash Function]. Moscow, Standartinform, 2012. 25 p. (in Russian)
51. *Alashwali E. and Rasmussen K.* What's in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS. Cryptology ePrint Archive, 2019, Report 2019/1083. https://eprint.iacr.org/2019/1083.
52. *Kachalin I. F., Kuz'min A. S., Suslov E. A., et al.* Ob osnovnykh kontseptsiyakh kriptograficheskoy stoykosti [About main conceptions of cryptographic security]. Proc. XII All-Russian School-Colloquium on Stochastic Methods and VI All-Russian Symp. Appl. Industr. Math., Sochi, Dagomys, October 1–7, 2005, pp. 982–983. (in Russian)
53. *Los' A. B., Nesterenko A. Yu., and Rozhkov M. I.* Kriptograficheskie metody zashchity informatsii [Cryptographic methods of data security]. Moscow, Yurayt Publ., 2016. 473 p. (in Russian)
54. IETF. RFC 4303. IP Encapsulating Security Payload (ESP), 2005. https://datatracker.ietf.org/doc/html/rfc4303.
55. IETF. RFC 7296. Internet Key Exchange Protocol Version 2 (IKEv2), 2014. https://datatracker.ietf.org/doc/html/rfc7296.
56. *Krawczyk H.* SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie — Hellman and its use in the IKE protocols. LNCS, 2003, vol. 2729, pp. 400–425.
57. R 50.1.113-2016. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Kriptograficheskie algoritmy, soputstvuyushchie primeneniyu algoritmov elektronnoy tsifrovoy podpisi i funktsii kheshirovaniya [Information Technology. Cryptographic Data Security. Cryptographic Algorithms to Accompany the Usage of Digital Signature and Hash Function Algorithms]. Moscow, Standartinform, 2016. 28 p. (in Russian)

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.85 DOI 10.17223/20710410/56/5

ОБНАРУЖЕНИЕ АНОМАЛИЙ В СОСТАВНЫХ ОБЪЕКТАХ В ФОРМАТЕ JSON[1]

Е. А. Шляхтина*,**, Д. Ю. Гамаюнов*

*Московский государственный университет имени М. В. Ломоносова, г. Москва, Россия
**ООО «Солидсофт», г. Москва, Россия

E-mail: elena.shlyakhtina@solidwall.io, gamajun@seclab.cs.msu.su

Работа посвящена проблеме защиты от компьютерных атак современных веб-приложений и мобильных приложений с «облачной» серверной частью. Рассматривается задача обнаружения вредоносного содержимого в данных в формате JSON, который стал одним из самых распространённых способов сериализации и передачи объектов между клиентской и серверной частями приложений. Предложен метод построения эталонной модели для некоторой заданной коллекции JSON-объектов, на основе которой можно обнаруживать аномалии различных типов. Эталонная модель строится на основе моделей простых типов, входящих в состав объектов из коллекции, а также схемы, которая обобщает их структуру. Экспериментально исследован метод построения эталонной модели с использованием модификаций, затрагивающих структуру JSON-объекта, а также внедрения SQL-инъекций, инъекций команд ОС и JavaScript/HTML-инъекций. Проведён анализ статистической значимости предсказаний модели, измерено качество работы модели, определяемое коэффициентом корреляции Мэтьюса, на тестовых выборках, состоящих из данных, взятых из трафика реальных веб-приложений.

Ключевые слова: *безопасность веб-трафика, обнаружение аномалий, машинное обучение.*

ANOMALY DETECTION IN JSON STRUCTURED DATA

E. A. Shliakhtina*,**, D. Y. Gamayunov*

*Lomonosov Moscow State University, Moscow, Russia
**SolidSoft, LLC, Moscow, Russia

In this paper, we address the problem of intrusion detection for modern web applications and mobile applications with the cloud-based server side, using malicious content detection in JSON data, which is currently one of the most popular data serialization and exchange formats between client and server parts of an application. We propose a method for building a JSON model for the given set of JSON objects capable of detection of structure and type anomalies. The model is based on the models for basic data types inside JSON collection objects and schema model that generalizes objects' structure in the collection. We performed experiments using modifications of objects'

structures and insertions of code injection attack vectors such as SQL injections, OS command injections, and JavaScript/HTML injections. The analysis showed statistical significance between the model's predictions and the presence of anomalies in the data gathered from the real web applications' traffic. The quality of the model's predictions was measured using the Matthews correlation coefficient (MCC). The MCC values computed on the data were close to one which indicates the model's high efficiency in solving the problem of anomaly detection in JSON objects.

Keywords: *web traffic security, anomaly detection, machine learning*

Введение

В настоящее время одним из самых распространённых способов сериализации и обмена данными между клиентской и серверной частями приложений является формат данных JavaScript Object Notation (JSON) — он используется как в динамических веб-приложениях, так и для обмена между мобильным клиентом и «облачной» серверной частью приложения, а также для реализации внешних API-приложений. При этом различного рода вредоносная активность и компьютерные атаки, которые эксплуатируют некоторую уязвимость в приложении, также используют формат JSON для взаимодействия с серверной частью приложения — например, автоматизированная переборная активность для некоторого API будет заключаться в отправке запросов с перебираемым значением некоторого параметра внутри JSON-объекта, а поиск уязвимостей внедрения кода или выхода из контекста данных в контекст команд злоумышленник будет в том числе реализовывать подстановкой атакующих векторов в поля значений в JSON-объектах. Таким образом, можно пытаться обнаружить такие атаки за счёт обнаружения аномалий в структуре или типах данных в JSON-объектах внутри HTTP-запросов к приложению. Так возникает задача построения эталонной модели для коллекции JSON-объектов с целью обнаружения недопустимого вредоносного содержимого в данных такого типа. Впоследствии такая модель может быть использована в системах защиты приложений и автоматизированного API для контроля потоков данных в формате JSON.

Типичным средством защиты веб-приложений от компьютерных атак в настоящее время является межсетевой экран WAF (Web Application Firewall), работающий на прикладном уровне и анализирующий HTTP-трафик. WAF располагается перед защищаемым веб-сервером и работает в режиме обратного прокси-сервера. Просматривая пакеты информации в трафике, WAF строит эталонную модель данных, что в режиме анализа позволяет ему принимать решение о допустимости анализируемого трафика, который может содержать различные атаки на уровне приложений. Существует и другой подход для обнаружения вредоносного содержимого в трафике — это распознавание атак на основе определённых паттернов или сигнатур, однако существенным недостатком таких систем является то, что ранее неизвестные атаки могут быть найдены с меньшей вероятностью, в отличие от систем обнаружения аномалий.

Данная работа посвящена построению модели, определяющей наличие аномального содержимого в данных в формате JSON. В её основе лежат модели простых типов, входящих в состав JSON-объектов из коллекции, а также схема JSON-объектов, которая описывает структуру всех объектов коллекции.

1. Постановка задачи

Цель данной работы — разработка модели для коллекции составных объектов в формате JSON, определяющей класс («аномальный», «нормальный»), к которому принадлежит объект, лучше, чем случайное гадание.

Объект считается аномальным, если содержит одну из следующих атак, которые подробно описаны в п. 2:

1) изменение имени JSON-атрибута;
2) добавление JSON-атрибута;
3) удаление JSON-атрибута;
4) изменение значения JSON-атрибута: внедрение SQL-инъекции;
5) изменение значения JSON-атрибута: внедрение инъекции команд ОС;
6) изменение значения JSON-атрибута: внедрение JavaScript/HTML-инъекции.

Условия, в рамках которых проводится исследование:

1) рассматриваются синтаксически корректные JSON-объекты, как аномальные, так и нормальные;
2) аномальный JSON-объект содержит только одну атаку из списка выше;
3) все данные до их модификации считаются нормальными.

Сформулируем нулевую и альтернативную гипотезы:

— $\mathbf{H_0}$: предсказание модели об аномальности объекта из JSON-коллекции и его истинный класс являются независимыми при любых выбранных атаках из списка выше, их носителях из коллекции и самой коллекции;
— $\mathbf{H_1}$: предсказание модели об аномальности объекта из JSON-коллекции и его истинный класс не являются независимыми хотя бы при некоторых выбранных атаках из списка выше, их носителях из коллекции и самой коллекции.

Выберем уровень значимости $\alpha = 0{,}01$ и в соответствии с ним в результате исследования с использованием критерия хи-квадрат примем решение о принятии или отклонении нулевой гипотезы.

2. Основные понятия

2.1. JSON-формат

JSON — это текстовый формат данных, основанный на синтаксисе объекта JavaScript. Он часто используется для передачи данных в веб-приложениях, поскольку многие среды программирования имеют функционал для работы с ним.

Данные в формате JSON представляют собой один из следующих объектов:

— record (запись) — набор пар (атрибутов) *ключ* : *значение*, разделённых запятыми, и заключенный в фигурные скобки ({ });
— array (массив) — упорядоченный набор значений, разделённых запятыми, и заключенный в прямые скобки ([ ]).

Составными типами будем называть record и array.

Простыми типами будем называть:

— string (строка) — заключённое в двойные кавычки упорядоченное множество символов юникода;
— number (число) — целое или вещественное число;
— boolean (булевский тип) — принимает значения `true`/`false`;
— null — ключевое слово в JavaScript, которое обозначает отсутствующий объект.

Типом ключа может быть только string, а типом значения — любой (составной или простой). В листинге 1 приведён пример JSON-объекта.

```
{
    "string": "Hello World!",
    "number": 12.05,
    "boolean": true,
    "record": {
        "number": 2,
        "array": [true, 7, null]
    }
}
```

Листинг 1. Пример данных в формате JSON

2.2. JSON-схема

JSON-схема — это объект в формате JSON, описывающий типы данных и структуру некоторого JSON-объекта. Схема может быть использована для валидации структуры данных, обрабатываемых на веб-сервере. Существуют различные стандарты описания структуры JSON-объекта, например JSON Schema [1]. В листинге 3 приведён пример схемы этого стандарта для JSON-объекта листинга 2.

```
{
    "name": "Tom",
    "age": 33,
    "kids": ["Sam", "Tony"]
}
```

Листинг 2. Пример данных в формате JSON

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "$id": "/example/",
    "type": "object",
    "required": [
        "name",
        "age",
        "kids"
    ],
    "properties": {
        "name": {
            "$id": "#name",
            "type": "string"
        },
        "age": {
            "$id": "#age",
            "type": "number"
        },
        "kids": {
            "$id": "#kids",
            "type": "array",
            "items": {
```

```
23                  "$id": "#items",
24                  "anyOf": [
25                      {
26                          "$id": "#0",
27                          "type": "string"
28                      }
29                  ]
30              }
31          }
32      },
33      "additionalProperties": true
34  }
```

Листинг 3. Пример схемы JSON-объекта листинга 2

Схема фиксирует структуру объекта и ограничивает типы данных для значений, соответствующих конкретным ключам, могут быть указаны также дополнительные сведения, например максимальное и минимальное допустимые значения типа number или обязательность/опциональность ключей в записи объекта, всё это зависит от выбора алгоритма построения схемы.

2.3. Атаки на веб-приложения

Рассмотрим ряд атак на веб-приложения, векторы которых могут быть встроены в JSON-объект, попадающий на веб-сервер. В рейтинге самых распространённых уязвимостей веб-приложений «OWASP Top 10 2021» сообщества Open Web Application Security Project (OWASP) инъекционные атаки входят в первую тройку. Главной причиной уязвимости, связанной с использованием инъекций, является отсутствие проверки данных, полученных и используемых веб-приложением, что ведёт к необходимости валидации данных.

SQL-инъекция

Structured Query Language (SQL) — это специальный язык, используемый в большинстве реляционных баз данных для запросов к ним, обработки данных и контроля доступа. SQL-инъекция — это атака, при которой в процессе конструировния SQL-выражения на веб-сервере, благодаря особенному пользовательскому вводу, полностью меняется логика выполнения SQL-запроса. При недостаточной проверке данных от пользователя злоумышленник может внедрить в форму веб-интерфейса приложения специальный код, содержащий кусок SQL-запроса. Такая манипуляция позволяет злоумышленнику исполнить произвольный запрос к базе данных и получить незаконный доступ к данным, хранящимся в этой базе.

Инъекция команд ОС

Инъекция команд ОС (Command Injection) — это атака, целью которой является выполнение произвольных команд операционной системы с помощью уязвимого веб-приложения. Например, если при разработке веб-приложения существует функционал, позволяющий пользователю взаимодействовать с ОС веб-сервера, то злоумышленнику достаточно сформировать ввод, содержащий команды ОС, которые в итоге будут выполняться с привилегиями уязвимого приложения.

JavaScript/HTML-инъекция

JavaScript/HTML-инъекция, также известная как Cross Site Scripting (XSS), — это атака, при которой вредоносный код встраивается в выдаваемую веб-системой стра-

ницу и исполняется в браузере пользователя. Этот код обычно является JavaScript, HTML или любым другим кодом, который может быть исполнен в браузере. XSS позволяет злоумышленникам перехватывать пользовательские сессии, подменять веб-страницы или перенаправлять пользователей на вредоносные сайты.

3. Обзор методов

Приведём обзор существующих методов по автоматизации построения схемы JSON-объектов и обнаружению вредоносного содержимого в них.

В работах [2 – 4] предлагаются алгоритмы построения схем JSON-объектов. В [2, 3] используется схожий подход: для каждого объекта в коллекции генерируется схема, на основе полученных схем строится древовидная структура, которая хранит информацию о схемах объектов из коллекции. Информация — это атрибуты объектов и их типы, путь от корня объекта к атрибуту, число встреч атрибута, позвляющее делать выводы о его опциональности. В результате получается схема, содержащая в себе все возможные атрибуты, находящиеся в коллекции, с пометкой об их опциональности.

В [4] разработан язык для описания структуры JSON-объектов. Предлагается параметрический и допускающий параллельную работу способ построения схемы. Алгоритм на основе механизма map-reduce применяется к большой коллекции. Он основан на параллельном извлечении схемы для каждого элемента коллекции и объединении этих схем, причём на reduce-этапе происходит объединение только эквивалентных схем. Отношение эквивалентности (ER) является параметром алгоритма, регулирующим баланс между детальностью и компактностью полученной схемы. В работе подробно рассматриваются два вида ER: Kind-equivalence и Label-equivalence:

- Kind-equivalence:
  - простые типы: в случае идентичности сливаются в один, а разные добавляются в объединение;
  - составные типы: если ключи не совпадают, то они добавляются к новой записи и помечаются как опциональные, если же ключи совпадают, то они сливаются в один, а их типы рекурсивно обрабатываются.
- Label-equivalence: отличие этого подхода в том, что слияние двух схем происходит, только если множества ключей обеих схем совпадают, иначе схемы добавляются в объединение.

Вводится формальное описание типов данных и возможных операций над ними. Такой строгий подход позволяет авторам приводить формальные доказательства свойств полученного метода. Таким образом, на этапе map происходит генерация схемы для каждого JSON-объекта из коллекции, а на reduce — слияние полученных типов. Нужно отметить, что приведенные в [4] отношения эквивалентности, по мнению авторов, приемлемы для большинства задач. Ясно, что в случае, когда в коллекции содержатся максимально разнородные объекты, подход Kind-equivalence создаст компактную, человеко-читаемую схему, в отличие от Label-equivalence. Иначе, подход Label-equivalence может дать более точное описание объектов коллекции, так как учитывается совместная встречаемость ключей объектов. Можно заметить, что итоговая схема, получаемая в [2, 3], представляет собой подход Kind-equivalence работы [4].

В [5] описана модель, в основе которой лежит ансамбль случайных деревьев (Random Forest ensemble algorithm). Он агрегирует результаты работы классификаторов, выявляющих аномалии при помощи таких методов, как измерение энтропии Шеннона, анализ n-грамм строкового представления объекта и измерение его длины, измерение схожести структуры JSON-объектов. Утверждается, что для обнаружения

вредоносного контента разработанный метод использует подход выявления аномалий, а не поиск паттернов конкретных атак. Однако для построения пространства признаков для тренировочной и тестовой выборки используются n-граммы векторов атак, а именно для определения таких признаков, как общее число n-грамм векторов атак в объекте и их число в каждом простом типе этого объекта. Таким образом, для анализа некоторого JSON-объекта нужно строить пространство признаков и вычислять данные параметры, основываясь на некотором словаре n-грамм векторов рассматриваемых атак. Кроме того, анализ важности признаков, приведённый в [5], показал, что число n-грамм векторов атак находится на втором месте в рейтинге признаков, оказывающих наибольшее влияние на решение модели.

В данной работе мы ставим задачу разработать модель, которая определяла бы наличие аномального содержимого в объекте без какой-либо дополнительной информации о векторах атак.

4. Моделирование нормы для коллекции JSON-объектов

Приведём подробное описание предлагаемой модели. Она представляет собой совокупность классификаторов, а результат её работы — агрегация результатов работы классификаторов, среди которых:

1) модель схемы;
2) модель строкового представления объектов;
3) модель длины строкового представления объектов;
4) модели простых типов внутри объектов:
   - булевского типа;
   - численного типа;
   - строкового типа.

Каждая из этих моделей возвращает значение `true`, если аномалий нет, `false` — если есть. Результатом работы всей модели является конъюнкция результатов работы перечисленных моделей. Далее подробно опишем каждую из них.

4.1. Модель схемы

В качестве метода построения схемы JSON коллекции выбран Label-equivalence-подход [4]. Постановка задачи предполагает большой набор однородных данных для анализа, поэтому размер схемы даже при этом подходе не будет большим, в то же время учёт совместной встречаемости ключей даст возможность точнее представлять структуру ожидаемых на веб-сервере данных. Далее определим вид схемы JSON-объекта, приведём алгоритм её построения, алгоритм слияния двух схем и введём меру схожести двух схем.

Выработка схемы

Определим множества типов, которые будут использоваться при построении схемы SCHEMA_TYPES = {"string", "number", "boolean", "null", "record", "array", "union"}, и разобьём их на следующие группы:

- простые типы: BASIC_TYPES = {"string", "number", "boolean", "null"};
- составные типы:
  - RECORD_TYPE = "record";
  - ARRAY_TYPE = "array";
- тип, обозначающий объединение схем, которые не удалось слить в одну: UNION_TYPE = "union".

Будем строить схему объекта рекурсивно в процессе обхода этого объекта в глубину. Сначала определяется тип текущего объекта из множества SCHEMA_TYPES, далее в зависимости от типа строится конкретная структура (т. е. схема), описывающая текущий объект. Соответствие типа и его схемы представлено в табл. 1. Схема объекта листинга 2 находится в листинге 4.

Таблица 1

Вид схем различных типов

Тип	Схема	Описание

BASIC_TYPES	`{`
	`  "type": "number"`
	`}`
		В поле "type" записывается текущий тип

RECORD_TYPE	`{`
	`  "type": "record",`
	`  "properties": {`
	`  "prop_1": {...},`
	`  ...`
	`  "prop_k": {...}`
	`  }`
	`}`
		В поле "type" записывается текущий тип, а поле "properties" содержит список ключей в рассматриваемом объекте с соответствующими схемами их значений

ARRAY_TYPE	`{`
	`  "type": "array",`
	`  "items": {...}`
	`}`
		В поле "type" записывается текущий тип, а поле "items" содержит схему значений, содержащихся в массиве

UNION_TYPE	`{`
	`  "type": "union",`
	`  "subtypes": [...]`
	`}`
		В поле "type" записывается текущий тип, а поле "subtypes" содержит список схем, которые не удалось слить в одну

```
{
    "type": "record",
    "properties": {
        "name": {
            "type": "string"
        },
        "age": {
            "type": "number"
        },
        "kids": {
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    }
}
```

Листинг 4. Пример схемы JSON-объекта листинга 2

Задача слияния двух схем в одну возникает, например, когда нужно построить схему значений массива, содержащего элементы разных типов. Согласно методу [4], возможно слияние только эквивалентных схем. При Label-equivalence-подходе эквивалентность схем означает:

1) совпадение значений в поле "type";
2) совпадение множества ключей в поле "properties" при наличии этого поля.

Приведём описание алгоритма слияния двух схем. Если две схемы, поданные на вход алгоритму, являются эквивалентными, то происходит их слияние, а именно:

— простые типы из BASIC_TYPES сливаются в один;
— две схемы типа RECORD_TYPE образуют новую схему типа RECORD_TYPE с одинаковым множеством ключей, значения которых далее рекурсивно обрабатываются алгоритмом слияния;
— две схемы типа ARRAY_TYPE образуют новую схему типа ARRAY_TYPE, где схема значений вырабатывается слиянием схем значений в этих массивах при помощи алгоритма слияния;
— две схемы типа UNION_TYPE дают в результате данного алгоритма схему типа UNION_TYPE, где схемы в составе поля "subtypes" максимально по возможности слиты.

Если же схемы не эквивалентны, то они объединяются в схему типа UNION_TYPE.

Построение схемы JSON-объекта происходит при обходе этого объекта в глубину. Схема для коллекции JSON-объектов может быть построена итеративно, где на каждом шаге строится схема текущего объекта и затем сливается со схемой, выработанной на основе предыдущих объектов коллекции.

Мера сходства схем

Схема, построенная для коллекции JSON-объектов, представляет собой ожидаемую структуру объектов. Необходимо задать меру, определяющую уровень сходства структуры анализируемого объекта с выработанной схемой. В [5] для измерения сходства структур двух JSON-объектов используется коэффициент Жаккара, принимающий значения из отрезка $[0, 1]$.

Определение 1. Коэффициент сходства Жаккара (Jaccard similarity coefficient) — бинарная мера сходства двух множеств A и B, определяемая формулой

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Если $A = \varnothing$ и $B = \varnothing$, то $J(A, B) = 1$.

Чтобы была возможность использовать этот коэффициент, нужно представить схему в виде некоторого множества. В [5] предлагается представлять объект как множество всех путей в нём, где путь — это упорядоченный набор ключей, ведущих к простому типу внутри объекта. Будем использовать тот же подход для представления схемы (которая по определению тоже JSON-объект) в виде множества путей в ней.

Множество путей для некоторой схемы есть набор строк фиксированной структуры: первая часть строки состоит из списка ключей, находящихся по данному пути, в квадратных скобках; затем подстрока `=>`; далее указывается значение, находящееся по данному пути.

Приведём пример измерения сходства двух схем. В листингах 6 и 8 представлены наборы путей, выделенных для схем из листингов 5 и 7 соответственно.

```
{
    "type": "record",
    "properties": {
        "name": {
            "type": "string"
        },
        "age": {
            "type": "number"
        }
    }
}
```

Листинг 5. Схема A

```
[
    "["type"] => "record"",
    "["properties"]["name"]["type"] => "string"",
    "["properties"]["age"]["type"] => "number""
]
```

Листинг 6. Множество путей в схеме A

```
{
    "type": "record",
    "properties": {
        "name": {
            "type": "string"
        },
        "age": {
            "type": "string"
        }
    }
}
```

Листинг 7. Схема B

```
[
    "["type"] => "record"",
    "["properties"]["name"]["type"] => "string"",
    "["properties"]["age"]["type"] => "string""
]
```

Листинг 8. Множество путей в схеме B

Тогда коэффициент сходства схем A и B запишется так:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{2}{4} = 0{,}5.$$

Однако использование такого подхода в чистом виде было бы некорректным, так как при построении схемы может быть использован тип UNION_TYPE, тогда в соответствующем поле "subtypes" находятся все возможные структуры, которые могут присутствовать в этом конкретном месте схемы одного объекта. Поэтому, чтобы

иметь возможность сравнивать пути в схеме анализируемого объекта и пути в схеме, построенной для коллекции, нужен алгоритм, который из любой схемы, содержащей тип UNION_TYPE, формирует список всех возможных вариантов схем без типа UNION_TYPE внутри и содержащихся в исходной схеме. То есть для формирования списка схем используются всевозможные сочетания подсхем, выбранных из элементов типа UNION_TYPE исходной схемы. Эти подсхемы и будут находиться на том месте, где в схеме коллекции присутствовал тип UNION_TYPE. В листинге 10 представлен результат применения этого алгоритма к схеме из листинга 9.

```
{
    "type": "record",
    "properties": {
        "a": { "type": "string" },
        "b": {
            "type": "union",
            "subtypes": [
                { "type": "string" },
                { "type": "number" }
            ]
        }
    }
}
```

Листинг 9. Схема с типом UNION_TYPE

```
{
    "type": "record",
    "properties": {
        "a": { "type": "string" },
        "b": { "type": "string" }
    }
}

{
    "type": "record",
    "properties": {
        "a": { "type": "string" },
        "b": { "type": "number" }
    }
}
```

Листинг 10. Простые схемы в составе схемы из листинга 9

Измерение сходства схемы объекта со схемой коллекции проводится следующим образом. Для каждой из схем формируется список всех возможных вариантов схем без типа UNION_TYPE внутри. Затем все полученные объекты в этих списках преобразуются в наборы путей по принципу, описанному выше. Теперь можно вычислять схожесть между наборами с использованием коэффициента Жаккара. Если схема объекта не содержит тип UNION_TYPE, то для неё подбирается максимально похожая, т. е. с максимальны коэффициентом Жаккара, схема из списка вариантов для схемы коллекции. Если схема объекта не содержит тип UNION_TYPE, то для каждого варианта схемы объекта подбирается максимально похожий вариант схемы коллекции,

а затем из полученного списка выбирается вариант, дающий наименьший коэффициент Жаккара. Такой подход обусловлен тем, что поставленная задача — это выявить отклонения в схеме объекта от нормы, выраженной схемой коллекции.

Модель схемы определяет соответствие или несоответствие данного JSON-объекта заданной схеме на основе введённой меры и её порогового значения, являющегося гиперпараметром модели. В случае соответствия схеме модель возвращает `true`, иначе — `false`.

4.2. Модель строкового представления объектов

В качестве модели строкового представления объекта используется статистическая модель, которая подстраивается под обучающую выборку из строк и вычисляет вероятность принадлежности новой строки к модели.

Пусть t — конкретная n-грамма, $n \in \{1, 2\}$. Вероятность того, что случайно взятая n-грамма строки имеет значение t, равна

$$\mathsf{P}[y = t] = \theta_t.$$

Воспользуемся байесовским подходом к оценке неизвестного параметра модели θ_t. В качестве априорного распределения θ_t возьмём стандартное равномерное, т. е. $U[0, 1]$, которое является частным случаем бета-распределения:

$$\mathbf{U}[0, 1] \equiv \mathbf{B}(1, 1).$$

С помощью формулы Байеса можно получить апостериорное распределение:

$$\mathsf{P}[\theta_t|Y] = \frac{\mathsf{P}[Y|\theta_t] \cdot \mathsf{P}[\theta_t]}{\mathsf{P}[Y]},$$

где Y — строка, представляющая собой обучающую выборку.

Пусть Y содержит N n-грамм и N_t из них есть t, тогда правдоподобие имеет распределение по Бернулли:

$$\mathsf{P}[Y|\theta_t] = \binom{N}{N_t} \theta_t^{N_t} (1 - \theta_t)^{N - N_t}.$$

Бета-распределение является сопряжённым априорным распределением для распределения Бернулли. Его использование в качестве априорного гарантирует, что апостериорное распределение тоже будет бета-распределением. Это существенно упрощает вычисление апостериорной вероятности простым добавлением количества появления и непоявления конкретной n-граммы t к существующим параметрам бета-распределения, а именно:

$$\mathsf{P}[\theta_t|Y] = \frac{\binom{N}{N_t} \theta_t^{N_t} (1 - \theta_t)^{N - N_t} \dfrac{\theta_t^{1-1} (1 - \theta_t)^{1-1}}{B(1, 1)}}{\displaystyle\int_0^1 \binom{N}{N_t} x^{N_t} (1 - x)^{N - N_t} \frac{x^{1-1} (1 - x)^{1-1}}{B(1, 1)}\, dx} = \frac{\theta_t^{N_t} (1 - \theta_t)^{N - N_t}}{\displaystyle\int_0^1 x^{N_t} (1 - x)^{N - N_t}\, dx} =$$

$$= \frac{\theta_t^{N_t} (1 - \theta_t)^{N - N_t}}{B(N_t + 1, N - N_t + 1)} \sim \mathbf{B}(N_t + 1, N - N_t + 1).$$

Взяв математическое ожидание по апостериорному распредению $p(\theta_t|Y)$, получим байесовскую оценку на параметр θ_t:

$$\int_0^1 \mathsf{P}[\theta_t|Y] \cdot \theta_t \, d\theta_t = \frac{N_t + 1}{N + 2}.$$

Описанный процесс можно итерационно повторять при появлении новых данных для дообучения модели. В таком случае текущее распределение параметра считается априорным и вычисляется новое апостериорное распределение.

Вероятность принадлежности новой строки y к модели можно рассчитать по формуле

$$\mathsf{P}[y|\Theta] = \prod_{t \in n\text{-grams}} \theta_t^{N_t}.$$

Рассчитанная по этой формуле вероятность для коротких строк будет иметь большее значение, чем для длинных, поэтому будем использовать следующую оценку на $\mathsf{P}[y|\Theta]$, которая получается из $\mathsf{P}[y|\Theta]$ взятием корня степени N и последующим логарифмированием:

$$\text{score} = \frac{\sum\limits_{t \in n\text{-grams}} N_t \, \log \theta_t}{N}.$$

Решение о принадлежности строки к модели выносится сравнением вычисленного для неё score с некоторым порогом. Предполагая, что оценки score распределены нормально, воспользуемся правилом трёх сигм, где mean — операция взятия выборочного среднего:

$$\mathsf{P}[\text{mean(scores)} - 3\sigma < \text{score} < \text{mean(scores)} + 3\sigma] \approx 0{,}997.$$

Для стандартного отклонения σ используется следующая оценка:

$$\sigma = \big(\text{mean(scores)} - \min(\text{scores})\big)/2.$$

Положим порог равным $\text{mean(scores)} - 3\sigma$. Для всех строк, для которых обе величины score (для одограмм и биграмм) выше данного порога, модель возвращает `true`, иначе — `false`.

4.3. Модель длины строкового представления объектов

Моделью длины строкового представления объекта является модель численного типа, которая в качестве данных использует последовательность длин строкового представления JSON-объектов из коллекции. Модель численного типа описана в п. 4.4.

4.4. Модели простых типов

Опишем модели типов "boolean", "number" и "string". Они позволяют определять соответствие наблюдаемого значения, находящегося по конкретному пути в объекте, ожидаемым значениям по такому пути. Примеры путей в объекте представлены в листинге 6. Для удобства работы с моделью будем хранить пары («путь», «модель простого типа»), что позволит быстро по пути отыскать соответствующую модель простого типа и осуществить проверку значения.

Модель булевского типа

В процессе обучения осуществляется подсчёт частот встречаемости значений `true` и `false`. Если частота значения, поданного на анализ, больше некторой границы, являющейся гиперпараметром модели, то значение считается нормальным и модель возвращает `true`, в противном случае — аномальным и модель возвращает `false`.

Модель численного типа

В процессе обучения определяется интервал, содержащий все значения из тренировочной выборки. Значение считается аномальным, если не попадает в этот интервал. В случае аномальности модель возвращает `false`, иначе `true`.

Модель строкового типа

В качестве модели строкового типа выступает одна из следующих моделей:

1) модель ограниченного множества значений;
2) модель целых чисел и чисел с плавающей точкой;
3) модель даты и времени;
4) модель номера банковской карты;
5) модель адреса электронной почты;
6) модель URL-адреса;
7) модель имени файла;
8) модель номера телефона;
9) модель идентификаторов (GUID, globally unique identifier);
10) модель строк из шестнадцатеричных цифр;
11) модель коротких строк.

Проверка строкового значения каждой из моделей означает проверку этого значения на соответствие формату строки, который отражён в названии модели. Идея подхода в том, чтобы выбрать более узкую модель из списка, если это возможно. В противном случае в качестве модели строкового типа используется статистическая модель, описанная в п. 4.2.

Имеющийся набор строк делится на две группы: тренировочный (20 %) и валидационный (80 %). В порядке, указанном в списке, значения из тренировочной выборки проверяются на соответствие выбранной модели. У каждой из них определён гиперпараметр — максимально допустимая доля значений, которые не соответствуют модели. Если при проверке доля значений выборки, не подходящих модели, меньше максимально допустимой доли, то модель считается подходящей для выборки. Таким образом, с использованием тренировочной выборки создается список из подходящих моделей. Далее каждая из них проверяется на валидационной выборке. В качестве модели строкового типа выбирается первая из подходящих моделей, прошедшая валидационную проверку.

5. Методика экспериментального исследования

Коллекции JSON-объектов, используемых для тестирования, были собраны из трафика реальных веб-приложений. Информация об используемых для тестирования наборах данных представлена в табл. 2.

Для обучения модели используется 60 % данных, для тестирования — 40 %. Опишем формирование тестовых наборов. Для каждой атаки, рассматриваемой в работе, формируются отдельные тестовые наборы. Процент модифицированных данных составляет от 5 до 95 % тестовой выборки с шагом 5 %. В зависимости от атаки модификация JSON-объекта означает:

1) изменение имени JSON-атрибута: у случайно выбранного атрибута изменяется имя добавлением к нему строки «_CHANGED»;
2) добавление JSON-атрибута: по случайно выбранному пути размещается дополнительный атрибут {"new_key": "new_value"};

Таблица 2

Коллекции JSON-объектов

Имя коллекции	Информация о веб-приложении	Количество объектов
data_0	Онлайн-магазин бытовой техники и электроники	1436
data_1	Онлайн-сервис для покупки авиабилетов	589
data_2	Онлайн-магазин товаров для дома и садоводства	2689
data_3	Онлайн-сервис для покупки туров	1000
data_4	Онлайн-сервис для покупки туров	1000
data_5	Онлайн-сервис для продажи билетов на поезда и самолеты	1431
data_6	Веб-приложение для покупки авиабилетов	7508

3) удаление JSON-атрибута: удаляется случайно выбранный атрибут из списка обязательных; если в коллекции таких нет, то удаляется любой случайно выбранный атрибут;
4) изменение значения JSON-атрибута (внедрение SQL-инъекции, команд ОС, JavaScript/HTML-инъекции): случайно выбирается атрибут и в его значение внедряется вектор атаки, причём если значение не строкового типа, то происходит полное замещение, в случае строки вектор атаки либо полностью замещает собой значение, либо внедряется в случайную позицию этой строки (этот выбор также происходит случайно).

Для случайного выбора используется функция choice из модуля random языка Python3. Векторы атак, используемые для тестирования, взяты из источников [6 – 9]. Их количество представлено в табл. 3.

Таблица 3

Количество векторов атак

Тип вектора атаки	Количество векторов
SQL-инъекция	12842
JavaScript/HTML-инъекции	12968
Инъекция команд ОС	558

Для оценки качества работы модели предлагается использовать коэффициент корреляции Мэтьюса (MCC) [10], который используется для измерения качества бинарных классификаций, особенно если размеры классов сильно различаются. Он определяется следующей формулой:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}.$$

В табл. 4 приведена матрица ошибок, где определены классы TN, FP, FN, TP.

Таблица 4

Матрица ошибок

Реальность	Предсказания модели	
	Есть аномалии (`false`)	Нет аномалий (`true`)
Есть аномалии (`false`)	TN	FP
Нет аномалий (`true`)	FN	TP

MCC принимает значение на интервале $[-1, 1]$. Значение 1 соответствует идеальному предсказанию, 0 — ситуации случайного предсказания, -1 — полностью противоположному предсказанию.

В работах [11, 12] показано, что данная мера в случае бинарной классификации является более информативной, чем F1-мера, Accuracy, Balanced accuracy, Bookmaker informedness, Markedness. Кроме того, существует связь между статистикой хи-квадрат и коэффициентом корреляции Мэтьюса для матрицы ошибок размера 2×2 [10]:

$$\chi^2 = N \cdot \mathrm{MCC}^2,$$

где N — объём тестовой выборки. Поэтому для проверки гипотезы H_0 достаточно сравнить MCC, рассчитанный для тестовой выборки, с критическим значением MCC, который соответствует заданному уровню значимости α. Для $\alpha = 0{,}01$ и одной степени свободы, как в нашем случае, $\chi^2 = 6{,}6349$. Значит, критическое значение MCC равно

$$\mathrm{MCC}_{\mathrm{crit}} = \pm\sqrt{\chi^2/N} = \pm\sqrt{66349/N}.$$

Если рассчитанный MCC удовлетворяет неравенству $|\mathrm{MCC}| \leqslant |\mathrm{MCC}_{\mathrm{crit}}|$, то гипотеза H_0 принимается, иначе — отклоняется.

6. Результаты экспериментального исследования

Приведём результаты тестирования предлагаемой модели со следующими гиперпараметрами:

— модель схемы: пороговое значение для меры сходства со схемой 0,99;
— модель булевского типа: пороговое значение для частоты встречаемости булевского значения 0,05;
— модель строкового типа: пороговое значение для доли несоответствующих модели значений 0,03.

При тестировании на имеющихся наборах данных получены значения MCC (рис. 1–7), где по оси абсцисс номерами отмечены следующие модификации:

1) изменение имени JSON-атрибута;
2) добавление JSON-атрибута;
3) удаление JSON-атрибута;
4) изменение значения JSON-атрибута: внедрение SQL-инъекции;
5) изменение значения JSON-атрибута: внедрение инъекции команд ОС;
6) изменение значения JSON-атрибута: внедрение JavaScript/HTML-инъекции.

Из диаграмм видно, что на всех проведённых тестах значение MCC значительно превысило критический MCC, а значит, гипотеза H_0 отклоняется, т. е. решения предлагаемой модели об аномальности объектов являются статистически значимыми.

На рис. 8 приведена диаграмма размаха MCC под названием «Experimental Group Posttest», полученная в [5]; тестирование проводилось с использованием тех же модификаций, что и в данной работе, с дополнительной модификацией — внедрением строк для переполнения буфера в значение JSON-атрибута. Процент модифицированных данных также менялся от 5 до 95 % с шагом 5 %.

Верхний квартиль MCC, полученный при тестировании модели из [5], при сравнении со всеми тестами предлагаемой модели находится ниже нижнего квартиля MCC, полученного нами на тестовых наборах данных, что говорит о лучшем качестве работы предлагаемой модели.

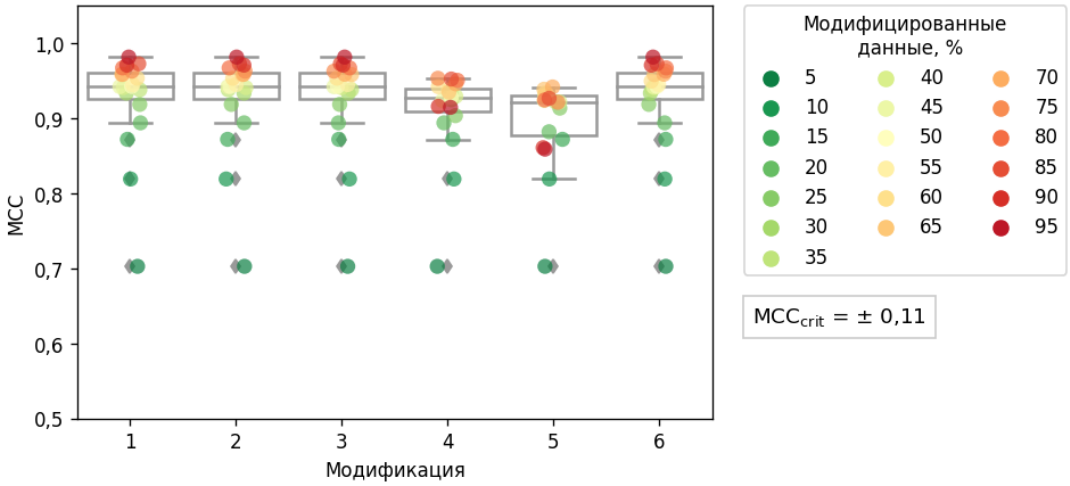

Рис. 1. Диаграмма размаха MCC для data_0

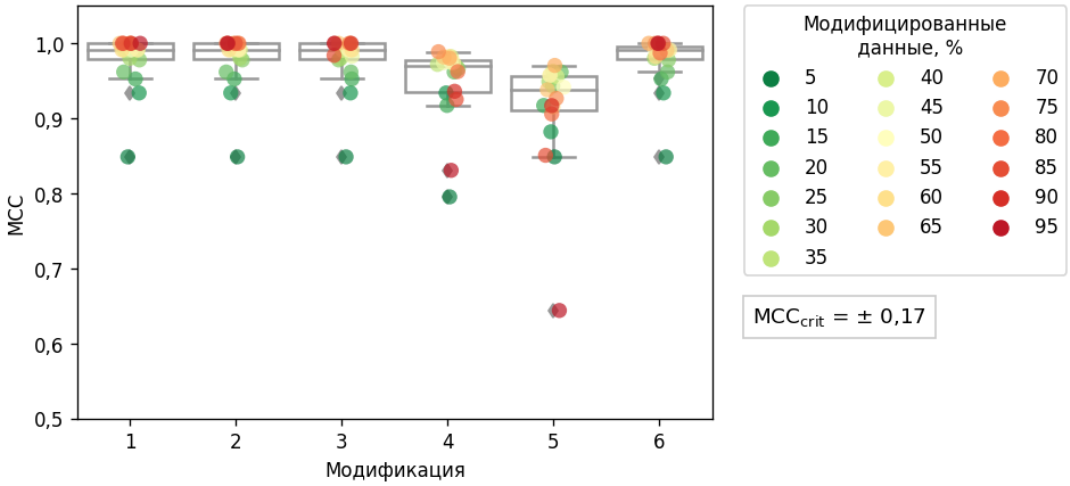

Рис. 2. Диаграмма размаха MCC для data_1

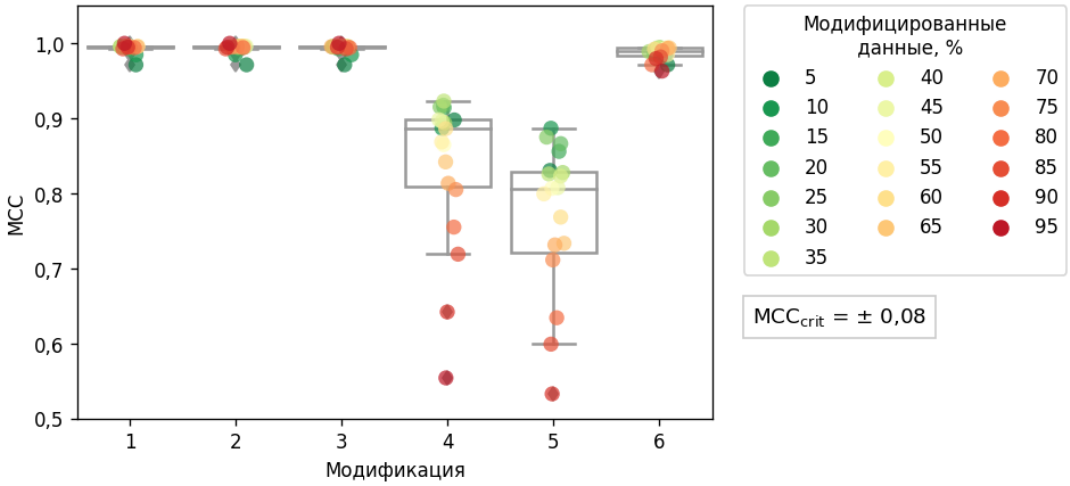

Рис. 3. Диаграмма размаха MCC для data_2

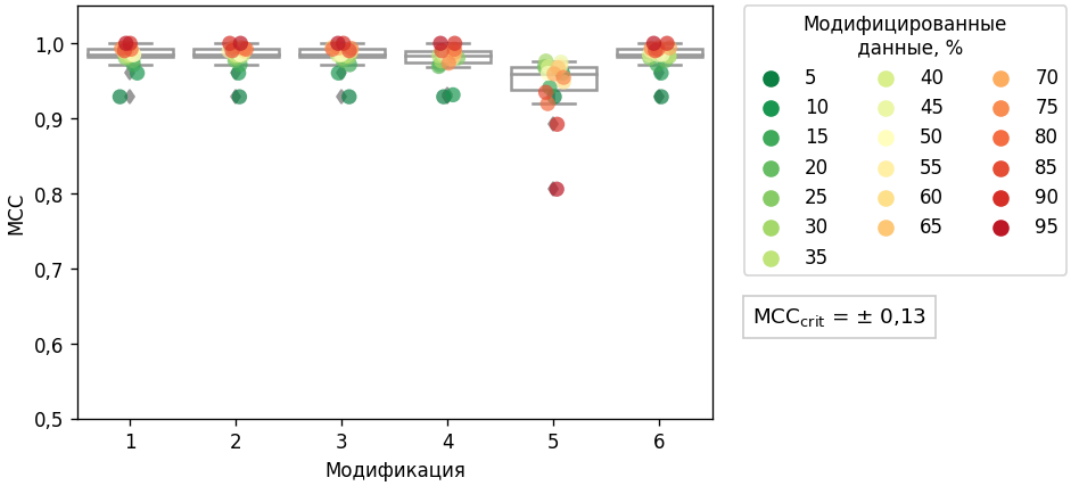

Рис. 4. Диаграмма размаха MCC для data_3

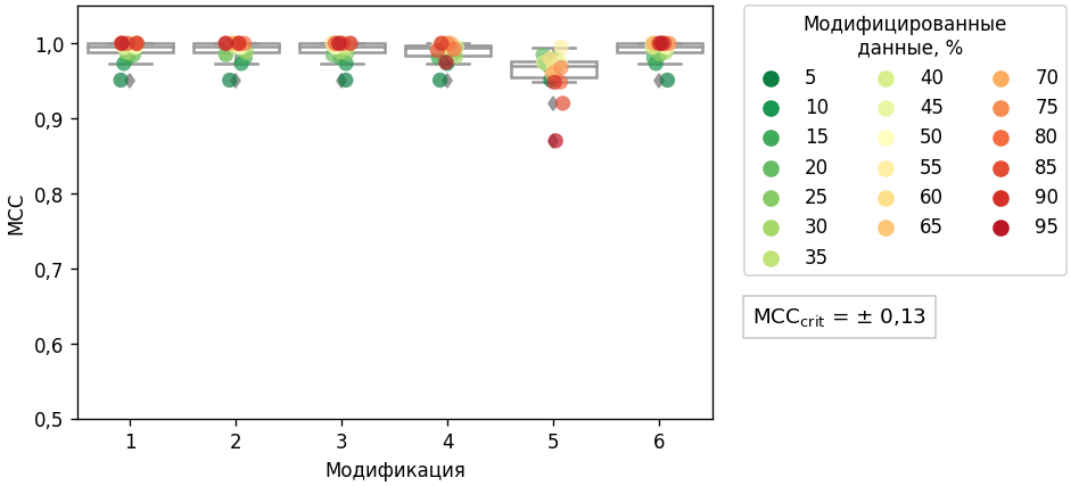

Рис. 5. Диаграмма размаха MCC для data_4

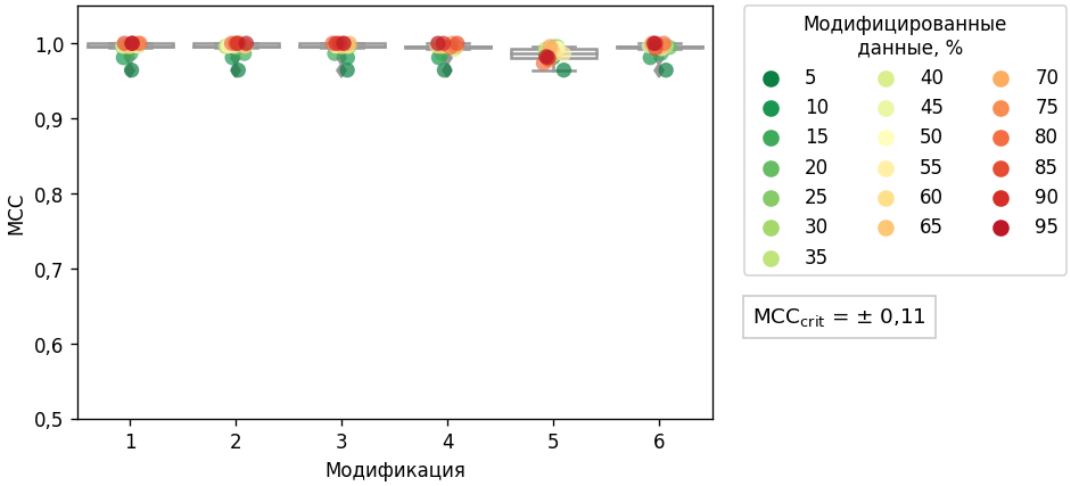

Рис. 6. Диаграмма размаха MCC для data_5

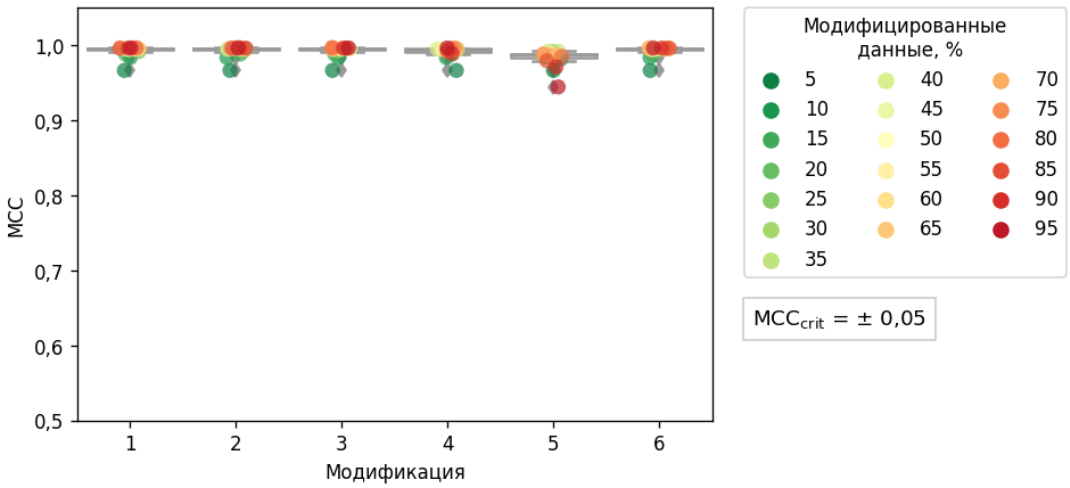

Рис. 7. Диаграмма размаха MCC для data_6

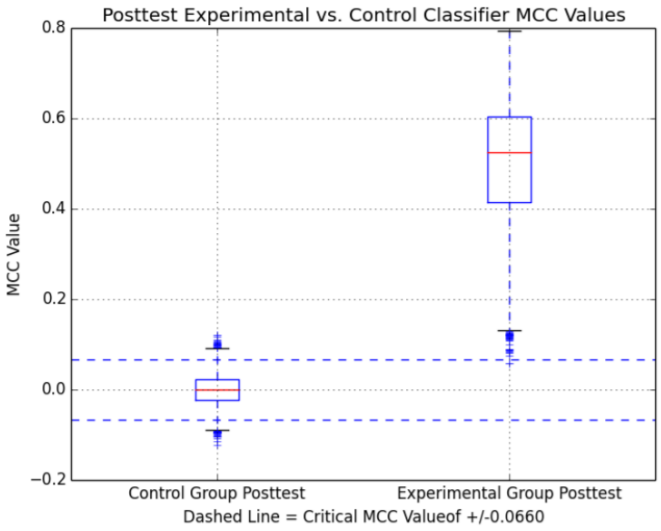

Рис. 8. Диаграмма размаха MCC, полученная в работе [5]

Сравнивая значения MCC при различных модификациях, можно сделать вывод, что для изменения имени JSON-атрибута, добавления JSON-атрибута, удаления JSON-атрибута модель показывает стабильно высокие значения, близкие к единице. Кроме того, с ростом процента модифицированных данных растёт MCC. Результаты модели при внедрении SQL-инъекции и JavaScript/HTML-инъекции также высокие, выбросы наблюдаются при минимальных долях модифицированных данных. Внедрение инъекции команд ОС показало средние значения MCC в сравнении с остальными модификациями, особенностью является то, что выбросы в этом случае относятся к данным с большим процентом модификации. Тем не менее медианы, полученные при всех экспериментах, выше 0,8, что говорит о стабильно хорошем качестве обнаружения аномалий моделью.

В табл. 5 представлено время проверки JSON-объекта из тестовых наборов данных, усреднённое по всем проведённым тестам для данной коллекции.

Таблица 5

Имя коллекции	Время, с
data_0	0,005461
data_1	0,000967
data_2	0,003866
data_3	0,002013
data_4	0,001439
data_5	0,000591
data_6	0,002125

Заключение

В работе рассмотрена проблема выявления вредоносного содержимого в составных объектах в формате JSON. Предложен метод построения модели, подстраивающейся под коллекцию JSON-объектов и позволяющей обнаружить аномальные, нетипичные для этой коллекции данные. Тестирование показало, что решения предлагаемой модели об аномальности объектов являются статистически значимыми. Значения MCC практически на всех тестовых данных близки к единице, что говорит о высокой результативности модели для решения задачи обнаружения структурных аномалий в JSON-объектах или аномалий нарушения типов данных для элементов JSON-объекта. Представленный подход может быть применён не только к формату JSON, но к любому типу данных, который может быть отображен в данный формат, например XML, который также часто используется для обмена информацией в веб-приложениях. Дальнейшие исследования могут быть связаны с разработкой дополнительных строковых моделей и расширением спектра возможных атак.

ЛИТЕРАТУРА

1. `www.json-schema.org`. JSON Schema. 2021.
2. *Frozza A. A., dos Santos Mello R., and da Costa F. S.* An approach for schema extraction of JSON and extended JSON document collections // IEEE Intern. Conf. IRI. 6–9 July 2018. P. 356–363.
3. *Klettke M., Störl U., and Scherzinger S.* Schema extraction and structural outlier detection for JSON-based NoSQL data stores // Conf. BTW, Hamburg, Germany, 4–6 March 2015. P. 425–444.
4. *Baazizi M. A., Colazzo D., Ghelli G., et al.* Parametric schema inference for massive JSON datasets // VLDB J. 2019. V. 28. No. 4. P. 497–521.
5. *Miller B. N.* Detection of Malicious Content in JSON Structured Data using Multiple Concurrent Anomaly Detection Methods. Dissertation. Eastern Michigan University, 2016. 125 p.
6. `www.github.com/payloadbox`. Payload Box. 2021.
7. `www.github.com/fuzzdb-project/fuzzdb`. FuzzDB Project. 2021.
8. `www.kaggle.com/syedsaqlainhussain/sql-injection-dataset`. SQL injection dataset. 2021.
9. `www.kaggle.com/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning`. Cross site scripting XSS dataset for Deep learning. 2021.
10. *Baldi P., Brunak S., Chauvin Y., et al.* Assessing the accuracy of prediction algorithms for classification: an overview // Bioinformatics. 2000. V. 16. No. 5. P. 412–424.

11. *Chicco D. and Jurman G.* The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation // BMC Genomics. 2020. V. 21. No. 1. P. 1–13.
12. *Chicco D., Totsch N., and Jurman G.* The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation // BioData Mining. 2021. V. 14. No. 1. P. 1–22.

REFERENCES

1. `www.json-schema.org`. JSON Schema, 2021.
2. *Frozza A. A., dos Santos Mello R., and da Costa F. S.* An approach for schema extraction of JSON and extended JSON document collections. IEEE Intern. Conf. IRI, 6–9 July 2018, pp. 356–363.
3. *Klettke M., Störl U., and Scherzinger S.* Schema extraction and structural outlier detection for JSON-based NoSQL data stores. Conf. BTW, Hamburg, Germany, 4–6 March 2015, pp. 425–444.
4. *Baazizi M. A., Colazzo D., Ghelli G., et al.* Parametric schema inference for massive JSON datasets. VLDB J., 2019, vol. 28, no. 4, pp. 497–521.
5. *Miller B. N.* Detection of Malicious Content in JSON Structured Data using Multiple Concurrent Anomaly Detection Methods. Dissertation. Eastern Michigan University, 2016. 125 p.
6. `www.github.com/payloadbox`. Payload Box. 2021.
7. `www.github.com/fuzzdb-project/fuzzdb`. FuzzDB Project. 2021.
8. `www.kaggle.com/syedsaqlainhussain/sql-injection-dataset`. SQL injection dataset. 2021.
9. `www.kaggle.com/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning`. Cross site scripting XSS dataset for Deep learning. 2021.
10. *Baldi P., Brunak S., Chauvin Y., et al.* Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 2000, vol. 16, no. 5, pp. 412–424.
11. *Chicco D. and Jurman G.* The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 2020, vol. 21, no. 1, pp. 1–13.
12. *Chicco D., Totsch N., and Jurman G.* The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Mining, 2021, vol. 14, no. 1, pp. 1–22.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ НАДЁЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

УДК 519.718.7 DOI 10.17223/20710410/56/6

КОРОТКИЕ ПОЛНЫЕ ДИАГНОСТИЧЕСКИЕ ТЕСТЫ ДЛЯ СХЕМ С ОДНИМ ДОПОЛНИТЕЛЬНЫМ ВХОДОМ В СТАНДАРТНОМ БАЗИСЕ[1]

К. А. Попков

Институт прикладной математики им. М. В. Келдыша РАН, г. Москва, Россия

E-mail: kirill-formulist@mail.ru

Доказано, что любую монотонную (антимонотонную) булеву функцию от n переменных можно смоделировать схемой из функциональных элементов с одним дополнительным входом в базисе «конъюнкция, дизъюнкция, отрицание», допускающей полный диагностический тест длины не более $n + 2$ (соответственно не более $n + 1$) относительно константных неисправностей типа 1 на выходах элементов.

Ключевые слова: *схема из функциональных элементов, константная неисправность, полный диагностический тест, булева функция.*

SHORT COMPLETE DIAGNOSTIC TESTS FOR CIRCUITS WITH ONE ADDITIONAL INPUT IN THE STANDARD BASIS

K. A. Popkov

Keldysh Institute of Applied Mathematics, Moscow, Russia

We prove that each monotone (antimonotone) Boolean function in n variables can be modeled by a logic circuit with one additional input in the basis "conjunction, disjunction, negation" allowing a complete diagnostic test with length no more than $n+2$ (no more than $n+1$, respectively) relative to constant faults of type 1 at outputs of logic gates.

Keywords: *logic circuit, stuck-at fault, complete diagnostic test, Boolean function.*

Введение

В работе рассматривается задача синтеза легкотестируемых схем из функциональных элементов (СФЭ), реализующих заданные булевы функции [1 – 3]. Пусть имеется СФЭ S с одним выходом, реализующая булеву функцию $f(\tilde{x}^n)$, где $n \in \mathbb{N}$ и $\tilde{x}^n = (x_1, \ldots, x_n)$. Предположим, что в схеме S могут происходить *константные неисправности типа a на выходах элементов*, при которых значение на выходе любого

[1]Работа выполнена при поддержке гранта РНФ, проект № 19-71-30004.

неисправного элемента становится равно заданной булевой константе a; число неисправных элементов в схеме предполагается произвольным. В результате схема S вместо исходной функции $f(\tilde{x}^n)$ будет реализовывать какие-то, вообще говоря, отличные от неё булевы функции, называемые *функциями неисправности* данной схемы.

Полным диагностическим тестом (ПДТ) для схемы S называется такое множество T наборов значений переменных $x_1, \ldots, x_n$, что для любой отличной от $f(\tilde{x}^n)$ функции неисправности $g(\tilde{x}^n)$ схемы S в T найдётся набор $\tilde{\sigma}$, на котором $f(\tilde{\sigma}) \neq g(\tilde{\sigma})$, а для любых двух различных функций неисправности $g_1(\tilde{x}^n)$ и $g_2(\tilde{x}^n)$ схемы S в T найдётся набор $\tilde{\pi}$, на котором $g_1(\tilde{\pi}) \neq g_2(\tilde{\pi})$. Число наборов в T называется *длиной* теста. В качестве тривиального ПДТ длины 2^n для схемы S всегда можно взять множество, состоящее из всех 2^n двоичных наборов длины n.

Пусть зафиксирован функционально полный базис B, в котором строятся схемы. Введём следующие обозначения: $D(T)$ — длина теста T; $D(S) = \min D(T)$, где минимум берётся по всем ПДТ T для схемы S; $D(f) = \min D(S)$, где минимум берётся по всем СФЭ S в базисе B, реализующим функцию f; $D(n) = \max D(f)$, где максимум берётся по всем булевым функциям f от n переменных. Функция $D(n)$ называется *функцией Шеннона* длины ПДТ.

Всюду далее будем считать, что n — произвольное натуральное число. В работе [4] для схем в базисе $\{x \,|\, y\}$, где $x \,|\, y = \overline{x\&y}$ — штрих Шеффера, в случае $a = 1$ установлена оценка $D(n) > \dfrac{2^{n/2} \cdot \sqrt[4]{n}}{2\sqrt{n + (\log n)/2 + 2}}$. Н. П. Редькин в [5] для схем в стандартном (классическом) базисе $\{\&, \vee, \neg\}$ в каждом из случаев $a = 0$, $a = 1$ получил оценку $D(n) \leqslant 2^{n-1}$, вдвое меньшую тривиальной оценки $D(n) \leqslant 2^n$.

Вместо «вход схемы S, отвечающий переменной x_i» для краткости будем писать «вход "x_i" схемы S».

Будем говорить, что СФЭ *моделирует* булеву функцию $f(\tilde{x}^n)$ *с n' дополнительными входами*, где $n' \in \mathbb{N}$, если данная схема реализует булеву функцию $\hat{f}(\tilde{x}^{n+n'})$, обладающую следующим свойством: существуют такие булевы константы $\sigma_1, \ldots, \sigma_{n'}$, что $f(\tilde{x}^n) = \hat{f}(x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_{n'})$.

Отметим, что моделирование булевых функций СФЭ с дополнительными входами с последующим тестированием этих схем легко осуществимо на практике: на входы «x_{n+1}», …, «$x_{n+n'}$» произвольной СФЭ, моделирующей булеву функцию $f(\tilde{x}^n)$ с n' дополнительными входами, в штатном режиме нужно подавать константы $\sigma_1, \ldots, \sigma_{n'}$ соответственно, а в режиме тестирования на каждый такой вход могут последовательно подаваться различные булевы значения.

1. Формулировка и доказательство основного результата

Булева функция $f(\tilde{x}^n)$ называется *монотонной* (*антимонотонной*), если для любых таких $\sigma_1, \ldots, \sigma_n, \pi_1, \ldots, \pi_n \in \{0, 1\}$, что $\sigma_1 \leqslant \pi_1, \ldots, \sigma_n \leqslant \pi_n$, выполняется неравенство $f(\sigma_1, \ldots, \sigma_n) \leqslant f(\pi_1, \ldots, \pi_n)$ (соответственно $f(\sigma_1, \ldots, \sigma_n) \geqslant f(\pi_1, \ldots, \pi_n)$). Очевидно, что если функция f монотонна, то функция $\overline{f}$ антимонотонна, и наоборот.

Рассмотрим стандартный базис $B_0 = \{\&, \vee, \neg\}$. Любой функциональный элемент, реализующий функцию вида $x\&y$ (вида $x \vee y$, вида $\overline{x}$), будем называть *конъюнктором* (соответственно *дизъюнктором*, *инвертором*).

Сформулируем основной результат данной работы.

Теорема 1. Справедливы следующие утверждения:

1) любую антимонотонную булеву функцию $f(\tilde{x}^n)$ можно смоделировать СФЭ с одним дополнительным входом в базисе B_0, допускающей ПДТ длины не более $n+1$ относительно константных неисправностей типа 1 на выходах элементов;

2) любую монотонную булеву функцию $f(\tilde{x}^n)$ можно смоделировать СФЭ с одним дополнительным входом в базисе B_0, допускающей ПДТ длины не более $n+2$ относительно константных неисправностей типа 1 на выходах элементов.

Доказательство.

1) Если $f \equiv \alpha$ для некоторого $\alpha \in \{0,1\}$, то указанную функцию можно смоделировать схемой с одним дополнительным входом «x_{n+1}», не содержащей функциональных элементов, выход которой совпадает с этим её входом (действительно, при подстановке вместо переменной x_{n+1} константы α реализуемая схемой функция становится равна f). Данная схема не имеет ни одной функции неисправности, поэтому допускает ПДТ длины $0 < n+1$, что и требовалось доказать.

Пусть функция $f(\tilde{x}^n)$ отлична от констант. По аналогии с [6, с. 45, лемма 5.2] легко доказать, что эту функцию можно представить в виде тупиковой дизъюнктивной нормальной формы

$$f(\tilde{x}^n) = K_1 \vee \ldots \vee K_m,$$

где $m \in \mathbb{N}$ и $K_1, \ldots, K_m$ — элементарные конъюнкции, в каждую из которых входят только отрицания переменных из множества $x_1, \ldots, x_n$. Домножим обе части последнего равенства на булеву переменную x_{n+1}:

$$\hat{f}(\tilde{x}^{n+1}) = x_{n+1} f(\tilde{x}^n) = K_1 x_{n+1} \vee \ldots \vee K_m x_{n+1}.$$

Пусть $K_i = \overline{x}_{j_1(i)} \ldots \overline{x}_{j_{r_i}(i)}$, где $r_i \in \{1, \ldots, n\}$ и $1 \leqslant j_1(i) < \ldots < j_{r_i}(i) \leqslant n$.

Построим схему S в базисе B_0, реализующую функцию $\hat{f}(\tilde{x}^{n+1})$ (рис. 1); в силу равенства $\hat{f}(x_1, \ldots, x_n, 1) = f(\tilde{x}^n)$ данная схема моделирует функцию $f(\tilde{x}^n)$ с одним дополнительным входом. Сначала для каждого $j = 1, \ldots, n$ подадим переменную x_j на вход инвертора I_j и поставим её в соответствие этому инвертору, а его выход и вход «x_j» схемы соединим со входами «контролирующего» конъюнктора C_j. Очевидно, что на выходе элемента I_j реализуется функция $\overline{x}_j$, а на выходе элемента C_j — константа 0. Далее, для каждого $i = 1, \ldots, m$ имеем $K_i x_{n+1} = \overline{x}_{j_1(i)} \ldots \overline{x}_{j_{r_i}(i)} x_{n+1}$. Реализуем выражение $K_i x_{n+1}$ цепочкой из r_i конъюнкторов $E_{i,1}, \ldots, E_{i,r_i}$, занумерованных «сверху вниз», входы которой последовательно соединим с выходами инверторов $I_{j_1(i)}, \ldots, I_{j_{r_i}(i)}$ и со входом «x_{n+1}» схемы. При этом в случае $r_i = 1$ входы конъюнктора $E_{i,1}$ соединяются с выходом инвертора $I_{j_1(i)}$ и со входом «x_{n+1}» схемы; в случае $r_i \geqslant 2$ — с выходами инверторов $I_{j_1(i)}$ и $I_{j_2(i)}$, а для каждого $t = 2, \ldots, r_i$ один вход конъюнктора $E_{i,t}$ соединяется с выходом конъюнктора $E_{i,t-1}$, а другой — с выходом инвертора $I_{j_{t+1}(i)}$ при $t \leqslant r_i - 1$ либо со входом «x_{n+1}» схемы при $t = r_i$. Затем в случае $r_i \geqslant 2$ для каждого $t = 2, \ldots, r_i$ поставим в соответствие конъюнктору $E_{i,t-1}$ переменные $x_{j_1(i)}, \ldots, x_{j_t(i)}$ и для каждого $p = 1, \ldots, t$ добавим в схему «контролирующий» конъюнктор $C_{i,t-1,p}$, входы которого соединим с выходом конъюнктора $E_{i,t-1}$ и со входом «$x_{j_p(i)}$» схемы. Наконец, выходы всех конъюнкторов E_{i,r_i}, C_j и $C_{i,t-1,p}$, где $i = 1, \ldots, m$, $j = 1, \ldots, n$, $t = 2, \ldots, r_i$ (при $r_i \geqslant 2$) и $p = 1, \ldots, t$ (при $r_i \geqslant 2$), соединим со входами цепочки из дизъюнкторов, выход которой будем считать выходом схемы.

Докажем, что схема S реализует функцию $\hat{f}(\tilde{x}_{n+1})$. По построению на выходе каждого конъюнктора E_{i,r_i} реализуется функция $K_i x_{n+1}$, на выходе каждого конъюнктора C_j — константа 0, а входы каждого конъюнктора $C_{i,t-1,p}$ соединены с выходом конъ-

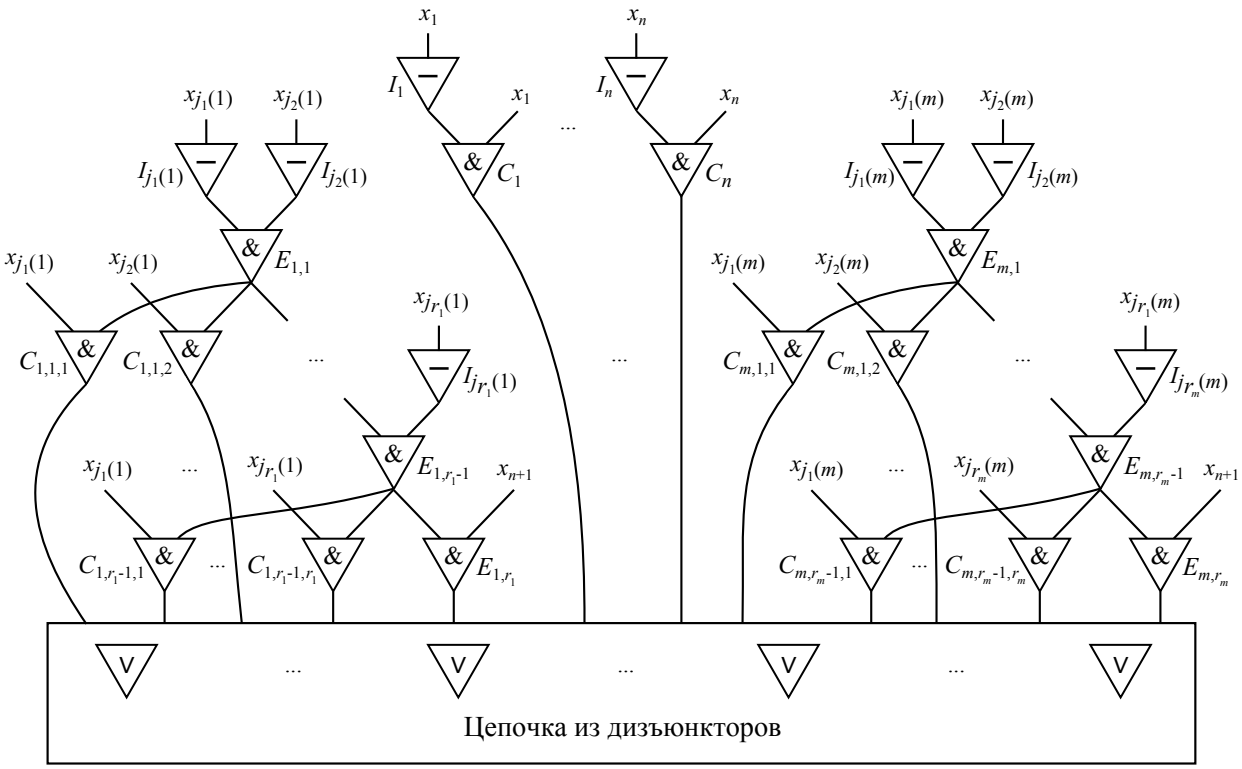

Рис. 1. Схема S

юнктора $E_{i,t-1}$ и со входом «$x_{j_p(i)}$» схемы S. При этом на выходе элемента $E_{i,t-1}$, как нетрудно видеть, реализуется «начало» $\overline{x}_{j_1(i)} \dots \overline{x}_{j_t(i)}$ элементарной конъюнкции K_i, следовательно, на выходе элемента $C_{i,t-1,p}$ реализуется функция $\overline{x}_{j_1(i)} \dots \overline{x}_{j_t(i)} x_{j_p(i)}$, тождественно равная нулю, так как $p \in \{1, \dots, t\}$. Таким образом, схема S реализует функцию

$$K_1 x_{n+1} \vee \dots \vee K_m x_{n+1} \vee 0 \vee \dots \vee 0 = \hat{f}(\tilde{x}^{n+1}).$$

Найдём все возможные функции неисправности схемы S. Если в ней неисправен хотя бы один дизъюнктор или хотя бы один из конъюнкторов E_{i,r_i}, C_j и $C_{i,t-1,p}$ (при всевозможных значениях индексов), то на входе или выходе какого-то дизъюнктора этой схемы возникнет константное значение 1, которое «пройдёт» по цепочке из дизъюнкторов до выхода схемы S; таким образом, схема будет реализовывать функцию $g_0(\tilde{x}^{n+1}) \equiv 1$. Пусть теперь все дизъюнкторы и конъюнкторы E_{i,r_i}, C_j, $C_{i,t-1,p}$ схемы S исправны, а хотя бы один из оставшихся её элементов неисправен. Тогда все неисправные элементы в схеме содержатся среди инверторов $I_1, \dots, I_n$ и конъюнкторов $E_{i,1}, \dots, E_{i,r_i-1}$ для каждого такого $i \in \{1, \dots, m\}$, что $r_i \geqslant 2$. Заметим, что каждому перечисленному в предыдущем предложении инвертору и конъюнктору по построению соответствует хотя бы одна переменная из множества $\{x_1, \dots, x_n\}$. Докажем, что схема S при указанной неисправности реализует функцию $g_{s_1,\dots,s_q}(\tilde{x}^{n+1}) = \hat{f}(\tilde{x}^{n+1}) \vee x_{s_1} \vee \dots \vee x_{s_q}$, где $x_{s_1}, \dots, x_{s_q}$ — все такие переменные, каждая из которых соответствует хотя бы одному неисправному элементу. Для этого достаточно доказать, что при подаче на входы схемы произвольного двоичного набора $(\sigma_1, \dots, \sigma_{n+1})$ на её выходе возникнет значение $\hat{f}(\sigma_1, \dots, \sigma_{n+1}) \vee \sigma_{s_1} \vee \dots \vee \sigma_{s_q}$. Рассмотрим два случая.

С л у ч а й 1. Существует такое $k \in \{1, \ldots, q\}$, что $\sigma_{s_k} = 1$. Тогда $\hat{f}(\sigma_1, \ldots, \sigma_{n+1}) \vee \vee \sigma_{s_1} \vee \ldots \vee \sigma_{s_q} = 1$. Переменная x_{s_k} соответствует хотя бы одному неисправному элементу. Возможны два подслучая.

1.1. Переменная x_{s_k} соответствует некоторому неисправному инвертору. Тогда это, очевидно, инвертор I_{s_k}, на его выходе реализуется константа 1, которая подаётся на один из входов конъюнктора C_{s_k}, а на другой вход этого конъюнктора подаётся переменная x_{s_k} со входа схемы, принимающая значение 1 на наборе $(\sigma_1, \ldots, \sigma_{n+1})$. Таким образом, на выходе элемента C_{s_k} на указанном наборе возникнет значение 1, которое подаётся на один из входов цепочки из дизъюнкторов и «пройдёт» до выхода схемы S.

1.2. Переменная x_{s_k} соответствует некоторому неисправному конъюнктору $E_{i,t-1}$. Тогда x_{s_k} — это одна из переменных $x_{j_1(i)}, \ldots, x_{j_t(i)}$, например $x_{j_u(i)}$. На выходе элемента $E_{i,t-1}$ реализуется константа 1, которая подаётся на один из входов конъюнктора $C_{i,t-1,u}$, а на другой вход этого конъюнктора подаётся переменная $x_{j_u(i)}$ со входа схемы, т. е. переменная x_{s_k}, принимающая значение 1 на наборе $(\sigma_1, \ldots, \sigma_{n+1})$. Таким образом, на выходе элемента $C_{i,t-1,u}$ на указанном наборе возникнет значение 1, которое подаётся на один из входов цепочки из дизъюнкторов и «пройдёт» до выхода схемы S.

С л у ч а й 2. Справедливы равенства $\sigma_{s_1} = \ldots = \sigma_{s_q} = 0$. Тогда при отсутствии неисправностей в схеме S на наборе $(\sigma_1, \ldots, \sigma_{n+1})$ на выходах инверторов $I_{s_1}, \ldots, I_{s_q}$ возникнут единицы, поэтому неисправности указанных инверторов никак не отразятся на значении, выдаваемом схемой на данном наборе. Далее, пусть $E_{i,t-1}$ — произвольный неисправный конъюнктор схемы S, для которого $2 \leqslant t \leqslant r_i$. Как показано выше, при отсутствии неисправностей в схеме S на выходе этого конъюнктора реализуется функция $\overline{x}_{j_1(i)} \ldots \overline{x}_{j_t(i)}$, причём ему соответствуют в точности переменные $x_{j_1(i)}, \ldots, x_{j_t(i)}$. Следовательно, каждая из переменных $x_{j_1(i)}, \ldots, x_{j_t(i)}$ является одной из переменных $x_{s_1}, \ldots, x_{s_q}$ и принимает на наборе $(\sigma_1, \ldots, \sigma_{n+1})$ значение 0, а тогда значение функции $\overline{x}_{j_1(i)} \ldots \overline{x}_{j_t(i)}$, т. е. значение, возникающее в схеме S на выходе конъюнктора $E_{i,t-1}$, на данном наборе равно $\overline{0}\& \ldots \&\overline{0} = 1$. Таким образом, неисправность элемента $E_{i,t-1}$ никак не отразится на значении, выдаваемом схемой на наборе $(\sigma_1, \ldots, \sigma_{n+1})$. В итоге получаем, что на выходе схемы S при подаче на её входы указанного набора возникнет «правильное» значение $\hat{f}(\sigma_1, \ldots, \sigma_{n+1})$, равное $\hat{f}(\sigma_1, \ldots, \sigma_{n+1}) \vee \sigma_{s_1} \vee \ldots \vee \sigma_{s_q}$. Случай 2 разобран.

Тем самым установлено, что схема S при рассматриваемой неисправности реализует функцию $g_{s_1,\ldots,s_q}(\tilde{x}^{n+1}) = \hat{f}(\tilde{x}^{n+1}) \vee x_{s_1} \vee \ldots \vee x_{s_q}$. Значит, каждая функция неисправности схемы S совпадает либо с функцией g_0, либо с функцией $g_{s_1,\ldots,s_q}$ для некоторых $q \in \{1, \ldots, n\}$ и $1 \leqslant s_1 < \ldots < s_q \leqslant n$. Докажем, что данная схема допускает ПДТ $\{\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_n\}$ (длины $n+1$), где $\tilde{\pi}_0 = (\underbrace{0, \ldots, 0}_{n+1})$, $\tilde{\pi}_j = (\underbrace{0, \ldots, 0}_{j-1}, 1, \underbrace{0, \ldots, 0}_{n+1-j})$ для каждого $j = 1, \ldots, n$. Заметим, что $\hat{f}(\tilde{\pi}_0) = \hat{f}(\tilde{\pi}_1) = \ldots = \hat{f}(\tilde{\pi}_n) = 0$, так как $\hat{f}(\tilde{x}^{n+1}) = x_{n+1} f(\tilde{x}^n)$, а $(n+1)$-я компонента каждого из наборов $\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_n$ равна 0. С учётом этого имеем

$$g_{s_1,\ldots,s_q}(\tilde{\pi}_j) = \begin{cases} 1, & \text{если } j \in \{s_1, \ldots, s_q\}, \\ 0, & \text{если } j \in \{0, 1, \ldots, n\} \setminus \{s_1, \ldots, s_q\}, \end{cases} \tag{1}$$

поскольку $g_{s_1,\ldots,s_q}(\tilde{x}^{n+1}) = \hat{f}(\tilde{x}^{n+1}) \vee x_{s_1} \vee \ldots \vee x_{s_q}$. Следовательно, для любых $q \in$ $\in \{1, \ldots, n\}$ и $1 \leqslant s_1 < \ldots < s_q \leqslant n$ функцию $g_{s_1,\ldots,s_q}$ можно отличить от функции $\hat{f}$ на

наборе $\tilde{\pi}_{s_1}$, а от функции $g_0 \equiv 1$ — на наборе $\tilde{\pi}_0$; функцию $\hat{f}$ можно отличить от функции g_0 на наборе $\tilde{\pi}_0$. Осталось отличить функцию $g_{s_1,\ldots,s_q}$ от каждой функции $g_{s'_1,\ldots,s'_{q'}}$, где $q' \in \{1,\ldots,n\}$, $1 \leqslant s'_1 < \ldots < s_{q'} \leqslant n$ и $\{s_1,\ldots,s_q\} \neq \{s'_1,\ldots,s'_{q'}\}$. Из последнего соотношения вытекает, что существует число j от 1 до n, принадлежащее ровно одному из множеств $\{s_1,\ldots,s_q\}$ и $\{s'_1,\ldots,s'_{q'}\}$. Тогда в силу (1) одна из функций $g_{s_1,\ldots,s_q}, g_{s'_1,\ldots,s'_{q'}}$ принимает на наборе $\tilde{\pi}_j$ значение 1, а другая — значение 0.

В итоге получаем, что схема S допускает ПДТ длины $n+1$. Утверждение 1 теоремы 1 доказано.

2) Случай, когда $f \equiv \alpha$ для некоторого $\alpha \in \{0,1\}$, рассматривается аналогично. Пусть функция $f(\tilde{x}^n)$ отлична от констант. Из монотонности этой функции следует антимонотонность функции $\overline{f}(\tilde{x}^n)$, которая при этом также является неконстантной. Из доказательства утверждения 1 вытекает, что функцию $x_{n+1}\overline{f}(\tilde{x}^n)$ можно реализовать схемой S' в базисе B_0, каждая функция неисправности которой совпадает либо с $g_0(\tilde{x}^{n+1}) \equiv 1$, либо с $g_{s_1,\ldots,s_q}(\tilde{x}^{n+1}) = x_{n+1}\overline{f}(\tilde{x}^n) \vee x_{s_1} \vee \ldots \vee x_{s_q}$ для некоторых $q \in \{1,\ldots,n\}$ и $1 \leqslant s_1 < \ldots < s_q \leqslant n$, причём множество $\{\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_n\}$, где $\tilde{\pi}_0 = (\underbrace{0,\ldots,0}_{n+1})$, $\tilde{\pi}_j = (\underbrace{0,\ldots,0}_{j-1}, 1, \underbrace{0,\ldots,0}_{n+1-j})$ для каждого $j = 1,\ldots,n$, является ПДТ для схемы S'.

Соединим выход схемы S' со входом инвертора I. Полученную схему обозначим через S; её выходом будем считать выход элемента I. Очевидно, что на этом выходе реализуется функция $\overline{x_{n+1}\overline{f}(\tilde{x}^n)} = \overline{x}_{n+1} \vee f(\tilde{x}^n)$. Данная функция при подстановке вместо переменной x_{n+1} значения 1 становится равной $f(\tilde{x}^n)$, поэтому схема S моделирует функцию $f(\tilde{x}^n)$ с одним дополнительным входом. При неисправности инвертора I данная схема будет реализовывать функцию $g_0(\tilde{x}^{n+1}) \equiv 1$. Если же инвертор I исправен, то каждая функция неисправности схемы S, очевидно, равна отрицанию некоторой функции неисправности схемы S'. Таким образом, все функции неисправности схемы S принадлежат множеству $\{g_0, \overline{g}_0, \overline{g}_{s_1,\ldots,s_q} : q = 1,\ldots,n;\ 1 \leqslant s_1 < \ldots < s_q \leqslant n\}$.

Пусть $\tilde{\rho}$ — произвольный такой двоичный набор длины n, что $f(\tilde{\rho}) = 0$, а $\tilde{\tau}$ — набор, получающийся из $\tilde{\rho}$ приписыванием справа $(n+1)$-й компоненты, равной единице. Докажем, что схема S допускает ПДТ $\{\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_n, \tilde{\tau}\}$ (длины $n+2$). Заметим, что любые две различные функции из множества $\{x_{n+1}\overline{f}, g_0, g_{s_1,\ldots,s_q} : q = 1, \ldots, n;\ 1 \leqslant s_1 < \ldots < s_q \leqslant n\}$, а значит, и любые две различные функции из множества $\{\overline{x_{n+1}\overline{f}}, \overline{g}_0, \overline{g}_{s_1,\ldots,s_q} : q = 1,\ldots,n;\ 1 \leqslant s_1 < \ldots < s_q \leqslant n\}$ можно отличить друг от друга хотя бы на одном наборе из множества $\{\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_n\}$, поскольку оно является ПДТ для схемы S'. Осталось отличить функцию $g_0 \equiv 1$ от каждой из функций $\overline{x_{n+1}\overline{f}}, \overline{g}_0, \overline{g}_{s_1,\ldots,s_q}$, где $q = 1,\ldots,n$ и $1 \leqslant s_1 < \ldots < s_q \leqslant n$, на наборах из множества $\{\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_n, \tilde{\tau}\}$, т. е. для каждой из функций $x_{n+1}\overline{f}, g_0, g_{s_1,\ldots,s_q}$ предъявить такой набор из этого множества, на котором она принимает значение 1. Из определения набора $\tilde{\tau}$ следует, что функция $x_{n+1}\overline{f}$ принимает на нём значение 1, а отсюда и из определения функции $g_{s_1,\ldots,s_q}$ — что $g_{s_1,\ldots,s_q}(\tilde{\tau}) = 1$ для любых $q \in \{1,\ldots,n\}$ и $1 \leqslant s_1 < \ldots < s_q \leqslant n$; кроме того, $g_0(\tilde{\tau}) = 1$, так как $g_0 \equiv 1$. Утверждение 2, а вместе с ним теорема 1 доказаны. ■

Используя принцип двойственности [7, с. 24], а также те соображения, что функция, двойственная к монотонной (антимонотонной) функции, монотонна (антимонотонна), а базис B_0 является двойственным самому себе, нетрудно установить, что утвержде-

ния 1 и 2 теоремы 1 остаются справедливыми при замене в их формулировках словосочетания «типа 1» на «типа 0».

2. Оценка числа функций $f(\tilde{x}^n)$, для которых $D(f) \leqslant n+1$

Из доказательства утверждения 1 теоремы 1 вытекает, что любую булеву функцию $\hat{f}(\tilde{x}^{n+1})$ вида $x_{n+1}f(\tilde{x}^n)$, где $f(\tilde{x}^n)$ — неконстантная антимонотонная булева функция, можно реализовать СФЭ в базисе B_0, допускающей ПДТ длины $n+1$ относительно константных неисправностей типа 1 на выходах элементов. Заменяя всюду в предыдущем предложении n на $n-1$ и меняя для удобства обозначения функций, получаем, что при $n \geqslant 2$ любая булева функция $f(\tilde{x}^n)$ вида $x_n f'(\tilde{x}^{n-1})$, где $f'(\tilde{x}^{n-1})$ — неконстантная антимонотонная булева функция, реализуема СФЭ в базисе B_0, допускающей ПДТ длины n (относительно указанных неисправностей), т. е. $D(f) \leqslant n$. Аналогично из доказательства утверждения 2 теоремы 1 можно получить, что при $n \geqslant 2$ любая булева функция $f(\tilde{x}^n)$ вида $\overline{x}_n \vee f''(\tilde{x}^{n-1})$, где $f''(\tilde{x}^{n-1})$ — неконстантная монотонная булева функция, реализуема СФЭ в базисе B_0, допускающей ПДТ длины $n+1$, т. е. $D(f) \leqslant n+1$.

Почти очевидно, что если булева функция $f_2(\tilde{x}^n)$ получается из булевой функции $f_1(\tilde{x}^n)$ перестановкой переменных, то $D(f_2) = D(f_1)$ (формальное доказательство этого факта см. в [8, с. 186, утверждение 8.10]). Рассмотрим множество F_n булевых функций от n переменных, состоящее из всех функций следующих видов:

$(*)$ $x_i f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, где $i \in \{1, \ldots, n\}$ и f' — неконстантная антимонотонная булева функция;

$(**)$ $\overline{x}_j \vee f''(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$, где $j \in \{1, \ldots, n\}$ и f'' — неконстантная монотонная булева функция.

В силу написанного выше $D(f) \leqslant n$ (соответственно $D(f) \leqslant n+1$) для любой булевой функции $f(\tilde{x}^n)$ вида $(*)$ (соответственно $(**)$). Таким образом, $D(f) \leqslant n+1$ для любой функции f из множества F_n.

Оценим снизу мощность данного множества. Прежде всего заметим, что никакая функция вида $(*)$ не совпадает ни с какой функцией вида $(**)$. Действительно, если $x_i f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = \overline{x}_j \vee f''(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$, то при подстановке $x_i = 0$ и $x_j = 0$ получаем противоречие $0 = 1$. Докажем, что никакие две функции вида $(*)$ при различных i не совпадают. Предположим противное: $x_{i_1} f_1'(x_1, \ldots, x_{i_1-1}, x_{i_1+1}, \ldots, x_n) = x_{i_2} f_2'(x_1, \ldots, x_{i_2-1}, x_{i_2+1}, \ldots, x_n)$, где $i_1 \neq i_2$. Тогда при подстановке $x_{i_1} = 0$ функция $x_{i_2} f_2'(x_1, \ldots, x_{i_2-1}, x_{i_2+1}, \ldots, x_n)$, а значит, и $f_2'(x_1, \ldots, x_{i_2-1}, x_{i_2+1}, \ldots, x_n)$, должна превратиться в тождественный нуль. Функция $f_2'(x_1, \ldots, x_{i_2-1}, x_{i_2+1}, \ldots, x_n)$ антимонотонна, поэтому при подстановке $x_{i_1} = 1$ она также должна стать тождественно нулевой, а тогда $f_2' \equiv 0$, что невозможно, так как функция f_2' отлична от констант. Похожим образом можно доказать, что никакие две функции вида $(**)$ при различных j не совпадают. В итоге имеем $|F_n| = nm'_{n-1} + nm''_{n-1}$, где m'_{n-1} — число неконстантных антимонотонных, а m''_{n-1} — число неконстантных монотонных булевых функций от $n-1$ переменных. Ясно, что $m'_{n-1} = m''_{n-1}$ (каждой монотонной булевой функции можно поставить во взаимно однозначное соответствие антимонотонную булеву функцию, являющуюся её отрицанием), поэтому

$$|F_n| = 2nm''_{n-1}. \tag{2}$$

Пусть $n \geqslant 3$. Любая булева функция от $n-1$ переменных, принимающая значение 0 на всех наборах с менее чем $\lfloor (n-1)/2 \rfloor$ единицами, значение 1 на всех наборах с более чем $\lfloor (n-1)/2 \rfloor$ единицами и произвольные значения на наборах ровно с $\lfloor (n-1)/2 \rfloor$

единицами, очевидно, является неконстантной и монотонной, откуда $m''_{n-1} \geqslant 2^{\mathrm{C}_{n-1}^{\lfloor (n-1)/2 \rfloor}}$. С учётом (2) имеем

$$|F_n| \geqslant n \cdot 2^{\mathrm{C}_{n-1}^{\lfloor (n-1)/2 \rfloor}+1}.$$

Используя (2) и асимптотику числа m''_n при $n \to \infty$, установленную в [9, теорема 1], можно найти асимптотику мощности множества F_n.

Пусть F_n^* — множество, состоящее из всех булевых функций, двойственных функциям из F_n. Тогда, очевидно, $|F_n^*| = |F_n|$, а из принципа двойственности легко вытекает, что $D(f) \leqslant n+1$ для любой функции f из множества F_n^* в случае $a = 0$, т. е. при рассмотрении константных неисправностей типа 0 на выходах элементов.

Заключение

Для каждого из случаев $a = 0$, $a = 1$ при $n \geqslant 3$ найдено по крайней мере $n \cdot 2^{\mathrm{C}_{n-1}^{\lfloor (n-1)/2 \rfloor}+1}$ булевых функций $f(\tilde{x}^n)$, удовлетворяющих неравенству $D(f) \leqslant n+1$ для схем в базисе $B_0 = \{\&, \vee, \neg\}$. Последнее неравенство существенно улучшает оценку $D(f) \leqslant 2^{n-1}$, вытекающую из неравенства $D(n) \leqslant 2^{n-1}$, установленного в [5].

Отметим также, что с учётом относительной простоты схемы S, построенной в ходе доказательства теоремы 1, и малой длины ПДТ для этой схемы результаты работы могут иметь практическое применение.

ЛИТЕРАТУРА

1. *Яблонский С. В.* Надёжность и контроль управляющих систем // Материалы Всесоюзного семинара по дискретной математике и её приложениям (Москва, 31 января–2 февраля 1984 г.). М.: Изд-во МГУ, 1986. С. 7–12.
2. *Яблонский С. В.* Некоторые вопросы надёжности и контроля управляющих систем // Математические вопросы кибернетики. Вып. 1. М.: Наука, 1988. С. 5–25.
3. *Редькин Н. П.* Надёжность и диагностика схем. М.: Изд-во МГУ, 1992. 192 с.
4. *Попков К. А.* Нижние оценки длин полных диагностических тестов для схем и входов схем // Прикладная дискретная математика. 2016. № 4(34). С. 65–73.
5. *Редькин Н. П.* К вопросу о длине диагностических тестов для схем // Математические заметки. 2017. Т. 102. Вып. 4. С. 624–627.
6. *Ложкин С. А.* Лекции по основам кибернетики (учебное пособие для студентов). М.: Издательский отдел ф-та ВМиК МГУ, 2004. 251 с.
7. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986. 384 с.
8. *Попков К. А.* О возможностях построения легкотестируемых контактных схем и схем из функциональных элементов: дис. ... д-ра физ.-мат. наук. М., 2021. 377 с.
9. *Коршунов А. Д.* О числе монотонных булевых функций // Проблемы кибернетики. Вып. 38. М.: Наука, 1981. С. 5–108.

REFERENCES

1. *Yablonskiy S. V.* Nadezhnost' i kontrol' upravlyayushchikh sistem [Reliability and verification of control systems]. Materialy Vsesoyuznogo seminara po diskretnoy matematike i ee prilozheniyam (Moscow, 31 Jan.–2 Feb. 1984). Moscow, MSU Publ., 1986, pp. 7–12. (in Russian)
2. *Yablonskiy S. V.* Nekotorye voprosy nadezhnosti i kontrolya upravlyayushchikh sistem [Some questions of reliability and verification of control systems]. Matematicheskie Voprosy Kibernetiki, iss. 1. Moscow, Nauka Publ., 1988, pp. 5–25. (in Russian)
3. *Red'kin N. P.* Nadezhnost' i diagnostika skhem [Circuits Reliability and Diagnostics]. Moscow, MSU Publ., 1992. 192 p. (in Russian)

4. *Popkov K. A.* Nizhnie otsenki dlin polnykh diagnosticheskikh testov dlya skhem i vkhodov skhem [Lower bounds for lengths of complete diagnostic tests for circuits and inputs of circuits]. Prikladnaya Diskretnaya Matematika, 2016, no. 4(34), pp. 65–73. (in Russian)
5. *Red'kin N. P.* Length of diagnostic tests for Boolean circuits. Math. Notes, 2017, vol. 102, iss. 3–4, pp. 580–582.
6. *Lozhkin S. A.* Lektsii po osnovam kibernetiki (uchebnoe posobie dlya studentov) [Lectures on the Basics of Cybernetics (Tutorial for Students)]. Moscow, MSU Faculty of Comput. Math. and Cybern. Publ., 2004. 251 p. (in Russian)
7. *Yablonskiy S. V.* Vvedenie v diskretnuyu matematiku [Introduction to Discrete Mathematics]. Moscow, Nauka Publ., 1986. 384 p. (in Russian)
8. *Popkov K. A.* O vozmozhnostyakh postroeniya legkotestiruemykh kontaknykh skhem i skhem iz funktsional'nykh elementov [On the Possibilities of Constructing Easily Testable Contact Circuits and Logic Circuits]: Diss. ... Doctor of Phys.-Math. Sci. Moscow, 2021. 377 p. (in Russian)
9. *Korshunov A. D.* O chisle monotonnykh bulevykh funktsiy [On the number of monotone Boolean functions]. Problemy Kibernetiki, iss. 38. Moscow, Nauka Publ., 1981, pp. 5–108. (in Russian)

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ДИСКРЕТНЫХ АВТОМАТОВ

УДК 519.7 DOI 10.17223/20710410/56/7

LOW POWER ASSIGNMENT OF PARTIAL STATES OF A PARALLEL AUTOMATON

Yu. V. Pottosin

United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus

E-mail: pott@newman.bas-net.by

The problem of a low-power assignment of the partial states of a parallel automaton is considered. A method to solve that problem is suggested that provides minimizing the number of memory elements in the implementing circuit of the automaton and minimization of their switching activity. The problem is reduced to finding a minimal weighted cover of a graph with its complete bipartite sub-graphs (bi-cliques).

Keywords: *parallel automaton, partial state, state assignment, complete bipartite sub-graph, weighted cover problem.*

ЭНЕРГОСБЕРЕГАЮЩЕЕ КОДИРОВАНИЕ ЧАСТИЧНЫХ СОСТОЯНИЙ ПАРАЛЛЕЛЬНОГО АВТОМАТА

Ю. В. Поттосин

Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь

Рассматривается задача кодирования частичных состояний параллельного автомата. Предложен метод решения, который обеспечивает минимизацию числа элементов памяти в схеме, реализующей автомат, и минимизацию интенсивности их переключений. Задача сводится к нахождению минимального взвешенного покрытия графа его полными двудольными подграфами (бикликами).

Ключевые слова: *параллельный автомат, частичное состояние, кодирование состояний, полный двудольный подграф, задача о взвешенном покрытии.*

1. Introduction

At present time, a great attention is paid to decreasing power consumption in designing discrete devices based on CMOS technology. It is caused by the tendency to increase the working time of power supply for portable devices and, on the other hand, by the tendency to lower acuity of the problem of heat rejection in designing VLSI circuits. Therefore, one of the main optimization criteria in designing discrete devices is amount of power consumption.

As it is said in [1, 2], the power consumption of a circuit built on the base of CMOS technology is proportional to switching activity of its logical and memory elements. It allows solving this problem at the level of logical design. In particular, decreasing power

consumption can be achieved in the stage of state assignment of an automaton when the states of a given automaton are assigned with binary or ternary vectors (codes) in order to obtain a system of Boolean functions necessary for constructing a logical net of a designed device. This problem was solved in [3 – 8] for classical sequential automata at synchronous and asynchronous realizations. The states of an automaton must be encoded in such a way that during the transition between its states as few memory elements as possible change their state. In this paper, a method for low-power assignment of partial states of a parallel automaton is suggested. The method uses the approach described in [9] that reduces the problem to the search for a cover of a graph with its complete bipartite sub-graphs (bicliques).

2. The used model

A parallel automaton is a functional model of a discrete device that gives a convenient description of the parallelism of controlled interactive processes [10]. This model is close to the widely known Petri net [11]. It consists of the following objects: a set of *partial states* $Q = \{q_1, q_2, \ldots, q_\gamma\}$, a set of input Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$, a set of output Boolean variables $Y = \{y_1, y_2, \ldots, y_m\}$ and a set of transitions $T = \{\tau_1, \tau_2, \ldots, \tau_t\}$ that is a sequence of lines of the form:

$$\tau_i = S_i : - \; K_i' \to K_i'' \to F_i, \tag{1}$$

where $S_i, F_i \subseteq Q$, K_i' is an elementary conjunction of variables from the set X, and K_i'' is an elementary conjunction of variables from the set Y.

Unlike the classical finite sequential automaton, the parallel automaton can be in several states simultaneously. Those states are called above as partial. The set of all partial states that a parallel automaton can be in at some time is called *global state*. The sense of the line (1) is the following. If a partial automaton is in states forming the set S_i and Boolean variables took values that convert K_i' to 1, then K_i'' takes value 1 and the automaton comes to states in F_i from states composing S_i. In other words, let $P = \{P_1, P_2, \ldots, P_p\}$ be the set of all reachable global states of a given parallel automaton. Then if $S_i \subseteq P_g$, where P_g is a current global state of the automaton, and the automaton has received binary signals that turned the conjunction K_i' into 1, then the global set will be $P_h = (P_g \setminus S_i) \cup F_i$ at the next time and the automaton will produce binary signals that turn K_i'' into 1. Any conjunction, K_i' and K_i'', can be absent in the line. The absence of K_i' means its identical equality to 1. The absence of K_i'' means, according to interpretation of the model, that either all the variables in Y are equal to 0 or the values of them do not change. As well as for a sequential automaton, the synchronous or asynchronous implementation can be for a parallel automaton. Further, the synchronous implementation is considered. At that implementation, the time is divided into fixed units, during which the automaton goes from one state to another.

The following restrictions are introduced in the model:

1) The initial global state is a one-element set. For the sake of determinacy, it can be $\{q_1\}$.
2) For two different lines, i-th and j-th, $S_i = S_j$ if $S_i \cap S_j \neq \varnothing$.

There are a number of other restrictions given in [10] and connected with correctness of an automaton description. They will not be considered here, since the correctness problem is not regarded. Also, the output signals will not be considered here.

Example 1. The following sequence of lines can be an example of parallel automaton:

$$\begin{array}{rcrl}
\tau_1 & = & 1: & -\ \overline{x}_1 x_2 \to 10; \\
\tau_2 & = & 10: & -\ \overline{x}_2 \to 2.3.4; \\
\tau_3 & = & 2: & \to 5.6; \\
\tau_4 & = & 3.5: & -\ x_2 \to 8; \\
\tau_5 & = & 4: & -\ \overline{x}_1 \to 7; \\
\tau_6 & = & 4: & -\ x_1 \to 9; \\
\tau_7 & = & 7: & -\ \overline{x}_2 \to 9; \\
\tau_8 & = & 6.8.9: & -\ x_1 \to 1.
\end{array}$$

Here, $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $X = \{x_1, x_2\}$. Let us take one-element set $\{1\}$ as an initial global state. The first line (transition τ_1) means that the automaton goes from state $\{1\}$ and comes in state $\{10\}$ in the next time unit if $x_1 = 0$ and $x_2 = 1$. The state $\{10\}$ is global one as well. The automaton stays in state $\{1\}$ at any other value combination of x_1 and x_2. The automaton goes from global state $\{10\}$ to partial states 2, 3 and 4 at $x_2 = 0$ (transition τ_2). Those partial states constitute the global state $\{2, 3, 4\}$. At the next time unit, the automaton changes the partial state 2 for partial states 5 and 6 independently on the values of input variables. As far as we consider the synchronous implementation, where transitions can happen simultaneously, the transitions τ_3 and τ_5 happen simultaneously at $x_1 = 0$, while τ_3 and τ_6 at $x_1 = 1$. Correspondingly, the automaton will be in global states $\{3, 5, 6, 7\}$ and $\{3, 5, 6, 9\}$. Having observed the functioning of the automaton in this way we get global states $\{6, 7, 8\}$ and $\{6, 8, 9\}$.

3. The problem of assignment of partial states

Any two partial states, in which the automaton can be simultaneously, are called *parallel*. The state assignment of an automaton consists in assigning its states with binary or ternary vectors of the space of introduced inner variables $z_1, z_2, \ldots, z_l$ (codes of states). The components of a ternary vector have values 0, 1 and "$-$". Parallel partial states of a parallel automaton are assigned with non-orthogonal ternary vectors and non-parallel partial states with orthogonal ones [12]. Two ternary vectors are orthogonal if there is a component that has value 0 in one vector and 1 in the other [13].

One of the way to establish parallelism between partial states of an parallel automaton is to obtain the set $P = \{P_1, P_2, \ldots, P_p\}$ of all reachable global states. Two partial states are parallel if there is a reachable global state $P_j \subseteq Q$ containing these states. The parallelism relation is considered to be irreflexive, i.e., a partial state is not parallel to itself. Indeed, the requirement of correctness of the automaton description does not allow intersection of sets $P_g \setminus S_i$ and F_i in transition τ_i, where P_g is the current global state of the automaton.

For the Example 1, the global states $P_1 = \{1\}$, $P_2 = \{10\}$, $P_3 = \{2, 3, 4\}$, $P_4 = \{3, 5, 6, 7\}$, $P_5 = \{3, 5, 6, 9\}$, $P_6 = \{6, 7, 8\}$, and $P_7 = \{6, 8, 9\}$ are obtained that yield the following matrix of partial states parallelism:

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\left[\begin{array}{cccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right] &
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10
\end{array}
\end{array}
$$

4. The method for partial state assignment

The suggested method for the assignment of partial states of parallel automaton supposes considering the non-parallelism graph G of partial states. Its vertices correspond to the partial states of a given automaton, and edges to the pairs of non-parallel partial states. Complete bipartite subgraphs (*bi-cliques*) are extracted from G. Each bi-clique B can be given enough by the pair $\langle V^1, V^2\rangle$ of its vertices, as every vertex in V^1 is connected by edges with all the vertices in V^2.

Let a family of bi-cliques $B_1, B_2, \ldots, B_m$ of graph G, where $B_i = \langle V_i^1, V_i^2\rangle$, $i = 1, 2, \ldots, m$, be a cover of G, i.e., for each edge of G there is a bi-clique in this family that has this edge. Every $B_k = \langle V_k^1, V_k^2\rangle$, $k = 1, 2, \ldots, m$, is put in correspondence to variable z_k with $z_k = 0$ for the partial states in V_k^1 and $z_k = 1$ for partial states in V_k^2 (or vice versa). The value of z_k is "$-$" if the corresponding partial state is neither in V_k^1 nor in V_k^2. It is natural to demand minimum of m. Having this minimum we obtain the assignment of partial states with codes of minimal length that relates to the minimum number of memory elements in the designed device. Evidently, it is enough to consider only maximal bi-cliques, i.e., any of them is not a proper subgraph of any other bi-clique.

Thus, the partial state assignment of a parallel automaton is reduced to the search for maximal bi-cliques in the non-orthogonality graph G of partial states of the given automaton and covering G with the found bi-cliques.

When applying this approach to lower the memory element activity, the *problem of minimal weighted cover* must be solved and the following reasons can be used.

Each inner variable z_i can be put into correspondence to a set of transitions between partial states. That set consists of the transitions which connect the partial states whose codes have different values of z_i. During these transitions the i-th memory element in a real circuit implementing the given automaton changes its state. Hence, in order to lower the activity of memory elements, such a variant of the assignment of partial states must be chosen that the number of variables changing their values during the transitions between states would be as small as possible.

If we manage to calculate the probabilities of transitions between partial states, then the probability of the transitions changing the value of z_i is put in correspondence to z_i. The more probability of the transition, the less inner variables have to change their values at that. We connect the transitions between partial states with the transitions between global states. Since the transitions between global states of a parallel automaton are incompatible events, the probability of the transition between partial states q_i and q_j is equal to the sum of

the probabilities of the transitions between those global states where there is exchange of q_i and q_j. When all the reachable global states of a given parallel automaton are determined, it is easy to construct the sequential automaton that is equivalent to the given one. Its states are the global states of the given parallel automaton. The Chapman — Kolmogorov method is used in [5] to calculate the probabilities of transitions between states of a sequential automaton, where the probabilities are found as a result of solving a system of linear equations with those probabilities as unknowns. This method can be applied only when the automaton is completely specified and its behavior graph is strongly connected directed graph. Otherwise, the number of transitions related to changing z_i, $i = 1, 2, \ldots, m$, can be put in correspondence to z_i.

Each maximal bi-clique of the non-parallelism graph G of partial states is weighted with a value proportional to the probability of transitions related to the corresponding inner variable. The weight of a cover is the sum of weights of bi-cliques constituting the cover.

Let us consider the parallel automaton from the Example 1. The non-parallelism matrix of partial states of it (the inverse of the parallelism matrix with retained main diagonal) that is the adjacency matrix of the non-orthogonality graph G of the partial state codes of the automaton is as follows:

$$\begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left[\begin{array}{cccccccccc} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}\right] \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \end{array}$$

A method for constructing all the maximal bi-cliques in a graph is described in [14]. The following maximal bi-cliques are obtained for graph G:

$B_1 = \langle\{1\}, \{2, 3, 4, 5, 6, 7, 8, 9, 10\}\rangle$,
$B_2 = \langle\{1, 2, 3, 4, 5\}, \{8, 10\}\rangle$,
$B_3 = \langle\{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{10\}\rangle$,
$B_4 = \langle\{1, 2, 3, 4, 5, 10\}, \{8\}\rangle$,
$B_5 = \langle\{1, 2, 4\}, \{5, 6, 7, 8, 9, 10\}\rangle$,
$B_6 = \langle\{1, 2, 4, 7\}, \{9, 10\}\rangle$,
$B_7 = \langle\{1, 2, 4, 7, 10\}, \{9\}\rangle$,
$B_8 = \langle\{1, 2, 4, 8\}, \{5, 10\}\rangle$,
$B_9 = \langle\{1, 2, 4, 8, 10\}, \{5\}\rangle$,
$B_{10} = \langle\{1, 2, 4, 9\}, \{7, 10\}\rangle$,
$B_{11} = \langle\{1, 2, 4, 9, 10\}, \{7\}\rangle$,
$B_{12} = \langle\{1, 2, 4, 10\}, \{5, 6, 7, 8, 9\}\rangle$,
$B_{13} = \langle\{1, 5\}, \{2, 4, 8, 10\}\rangle$,
$B_{14} = \langle\{1, 5, 6, 7, 8, 9\}, \{2, 4, 10\}\rangle$,
$B_{15} = \langle\{1, 5, 6, 7, 8, 9, 10\}, \{2, 4\}\rangle$,
$B_{16} = \langle\{1, 5, 10\}, \{2, 4, 8\}\rangle$,
$B_{17} = \langle\{1, 7\}, \{2, 4, 9, 10\}\rangle$,
$B_{18} = \langle\{1, 7, 10\}, \{2, 4, 9\}\rangle$,
$B_{19} = \langle\{1, 8\}, \{2, 3, 4, 5, 10\}\rangle$,
$B_{20} = \langle\{1, 8, 10\}, \{2, 3, 4, 5\}\rangle$,
$B_{21} = \langle\{1, 9\}, \{2, 4, 7, 10\}\rangle$,
$B_{22} = \langle\{1, 9, 10\}, \{2, 4, 7\}\rangle$,
$B_{23} = \langle\{1, 10\}, \{2, 3, 4, 5, 6, 7, 8, 9\}\rangle$.

5. Calculating probabilities of transitions between partial states and weighting bi-cliques

As it was said above, the probability of changing a partial state for another is equal to the sum of probabilities of transitions between global states, where this changing happens. Considering the sequential automaton equivalent to a given parallel automaton, whose global states in $P = \{P_1, P_2, \ldots, P_\gamma\}$ are in one-to-one correspondence to the states in $S = \{s_1, s_2, \ldots, s_\gamma\}$ of the sequential automaton, is appropriate here.

The probability of transition of a sequential automaton from a state s_i to a state s_j caused by an input signal $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is equal to the probability of coming $\mathbf{x}$. If there are several input signals transferring the automaton from s_i to s_j, the conditional probability p'_{ij} of such transfer is equal to the sum of the probabilities of those signals. The condition is that the automaton is in the state s_i. The absolute probability p_{ij} of the transition from s_i to s_j for all the time of the automaton working is equal to the product $p(s_i)p'_{ij}$, where $p(s_i)$ is the probability that the automaton is in the state s_i — this event and coming signals that transfer the automaton from s_i to s_j, are independent events. The following system of equations has to be solved:

$$\sum_{i=1}^{\gamma} p(s_i)p'_{ij} = p(s_j), \quad j = 1, \ldots, \gamma,$$

$$\sum_{i=1}^{\gamma} p(s_i) = 1.$$

The probability p'_{ij} must be known. This is the probability of coming an input signal that transfers the automaton from state s_i into s_j. Further, we assume that the probabilities of input signals are uniformly distributed. Having solved this system of equations we obtain the probabilities $p(s_i) = p(P_i)$, $i = 1, 2, \ldots, \gamma$.

The transitions between global states of the parallel automaton in the Example 1 are shown in Table 1, where rows and columns correspond to global states, and its entry is the condition of the transition from the global state corresponding to the row into the global state corresponding to the column.

Table 1

States	1	10	$\{2,3,4\}$	$\{3,5,6,7\}$	$\{3,5,6,9\}$	$\{6,7,8\}$	$\{6,8,9\}$
1	$x_1 \vee \overline{x}_2$	$\overline{x}_1 x_2$					
10		x_2	$\overline{x}_2$				
$\{2,3,4\}$				$\overline{x}_1$	x_1		
$\{3,5,6,7\}$					$\overline{x}_2$	x_2	
$\{3,5,6,9\}$					$\overline{x}_2$		x_2
$\{6,7,8\}$						x_2	$\overline{x}_2$
$\{6,8,9\}$	x_1						$\overline{x}_1$

For the probabilities of global states of the considered parallel automaton according to Table 1, the following system of equations is formed:

$$\begin{aligned}
p(1) &= 3/4\,p(1) + 1/2\,p(\{6,8,9\}),\\
p(10) &= 1/2\,p(10) + 1/4\,p(1),\\
p(\{2,3,4\}) &= 1/2\,p(10),\\
p(\{3,5,6,7\}) &= 1/2\,p(\{2,3,4\}),\\
p(\{3,5,6,9\}) &= 1/2\,p(\{3,5,6,9\}) + 1/2\,p(\{2,3,4\}) + 1/2\,p(\{3,5,6,7\}),\\
p(\{6,7,8\}) &= 1/2\,p(\{6,7,8\}) + 1/2\,p(\{3,5,6,7\}),\\
p(\{6,8,9\}) &= 1/2\,p(\{6,8,9\}) + 1/2\,p(\{3,5,6,9\}) + 1/2\,p(\{6,7,8\}),\\
p(1) + p(10) &+ p(\{2,3,4\}) + p(\{3,5,6,7\}) + p(\{3,5,6,9\}) + p(\{6,7,8\}) + p(\{6,8,9\}) = 1.
\end{aligned}$$

The solution of this system gives the probabilities $p(1) = 8/23$, $p(10) = 4/23$, $p(\{2,3,4\}) = 2/23$, $p(\{3,5,6,7\}) = 1/23$, $p(\{3,5,6,9\}) = 3/23$, $p(\{6,7,8\}) = 1/23$, $p(\{6,8,9\}) = 4/23$.

Table 2 represents the absolute probabilities of transitions between global states. It is similar to Table 1, but contains probabilities instead of transition conditions. Table 3 shows the probabilities of transitions between partial states of the parallel automaton under consideration. As it is said above, the probability of the transition between partial states q_i and q_j is equal to the sum of the probabilities of the transitions between those global states, where there is exchange of q_i and q_j, because the transitions between global states of a parallel automaton are incompatible events. For example, the automaton goes from partial state 2 into partial states 5 and 6 when it goes from global state $\{2,3,4\}$ to global state $\{3,5,6,7\}$ or $\{3,5,6,9\}$. Then the probabilities of transitions from 2 to 5 and from 2 to 6 equal $1/23 + 1/23 = 2/23$.

Table 2

States	1	10	$\{2,3,4\}$	$\{3,5,6,7\}$	$\{3,5,6,9\}$	$\{6,7,8\}$	$\{6,8,9\}$
1		2/23					
10			2/23				
$\{2,3,4\}$				1/23	1/23		
$\{3,5,6,7\}$					1/46	1/46	
$\{3,5,6,9\}$							3/46
$\{6,7,8\}$							1/46
$\{6,8,9\}$	2/23						

Table 3

States	1	2	3	4	5	6	7	8	9	10
1										2/23
2					2/23	2/23				
3								2/23		
4							1/23		1/23	
5								2/23		
6	2/23									
7	2/23								1/23	
8	2/23									
9										
10		2/23	2/23	2/23						

The edges of the non-orthogonality graph G of the partial state codes of the automaton correspond to the pairs of non-parallel partial states of the automaton. We weight the edges of G by the numbers proportional to the probabilities of transitions in the corresponding pairs of partial states. As these numbers, we take the numerators of the fractional probabilities with a common denominator.

6. Solving the minimal weighted cover problem

The method for solving the problem of minimal weighted cover is described in [15]. The table of covering graph G under consideration is given by Table 4, where some columns are removed according to the reduction rule: a column a can be excluded from consideration if it has ones everywhere a column b has ones [13]. The table cannot be represented completely because of restricted size of the page.

Table 4

Maximal bi-cliques	Edges of G														Weight
	(1,3)	(1,6)	(1,10)	(2,6)	(3,8)	(3,10)	(4,8)	(4,9)	(5,8)	(6,10)	(7,9)	(7,10)	(8,10)	(9,10)	
B_1	1	1	1												16
B_2			1		1	1	1		1						28
B_3			1			1				1		1	1	1	16
B_4					1		1		1				1		12
B_5		1	1	1			1	1							36
B_6			1					1			1	1			20
B_7								1			1			1	8
B_8			1						1				1		16
B_9									1						8
B_{10}			1								1			1	16
B_{11}											1	1			4
B_{12}		1		1			1	1		1		1	1	1	24
B_{13}			1						1						16
B_{14}			1	1			1			1		1	1	1	18
B_{15}				1			1	1							20
B_{16}									1				1		16
B_{17}			1								1	1			12
B_{18}														1	16
B_{19}	1		1		1										12
B_{20}	1				1	1	1		1						20
B_{21}			1					1			1			1	10
B_{22}								1			1	1			12
B_{23}	1	1				1				1		1	1	1	28

The family of bi-cliques $\{B_{12}, B_{20}, B_{21}\}$ is the solution with the weight 54 for Table 4. To appreciate the quality of the solution, let we do not take into consideration the weights of bi-cliques. In that case the solution would be $\{B_6, B_{12}, B_{20}\}$ with the weight 64. The matrices of coding constructed according to these families, where the rows represent the codes of the partial states, are as follows:

$$\begin{array}{ccccc} z_1 & z_2 & z_3 & & \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & - & - \\ 1 & 0 & 1 \\ 1 & 1 & - \\ - & 1 & - \\ - & 1 & 1 \\ 0 & 1 & - \\ - & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] & & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} & , & \begin{array}{ccc} z_1 & z_2 & z_3 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ - & - & 1 \\ 0 & 0 & 1 \\ - & 1 & 1 \\ - & 1 & - \\ 0 & 1 & - \\ - & 1 & 0 \\ 1 & 1 & - \\ 1 & 0 & 0 \end{array}\right] & & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \end{array} \end{array} .$$

The quality of the state assignment problem solution for a sequential automaton can be appreciated by the value $D = \sum p_{ij}(d_{ij} - 1)$, where p_{ij} is the probability of the transition between states s_i and s_j in both directions, d_{ij} is the Hamming distance between the codes of s_i and s_j $(i \neq j)$, and summation is made over all the pairs of states. This value was introduced in [5]. Evidently, the less value of D, the better solution, and $D = 0$ if any transition between states corresponds to switching only one memory element in the circuit implementing the automaton.

In the case of a parallel automaton, such an appreciation of the quality of state assignment can be used considering the global states. The code of a global state can be easily obtained by intersection of ternary vectors coding the partial states that constitute the global one. The result of intersection of non-orthogonal ternary vectors **u** and **v** is the vector **w** obtained in the following way: a component w_i has value "$-$" if both u_i and v_i have this value, and $w_i = 0$ (or 1) if at least one of u_i and v_i has this value. The codes of the global states of the parallel automaton under consideration are represented by the following matrices, where the left matrix is obtained taking into account the weights of bi-cliques, and the right one without taking into account the weights:

$$\begin{array}{cccc} z_1 & z_2 & z_3 & \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{array}\right. & & & \left.\begin{array}{l} 1 \\ 10 \\ \{2,3,4\} \\ \{3,5,6,7\} \\ \{3,5,6,9\} \\ \{6,7,8\} \\ \{6,8,9\} \end{array}\right] \end{array},$$

$$\begin{array}{cccc} z_1 & z_2 & z_3 & \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{array}\right. & & & \left.\begin{array}{l} 1 \\ 10 \\ \{2,3,4\} \\ \{3,5,6,7\} \\ \{3,5,6,9\} \\ \{6,7,8\} \\ \{6,8,9\} \end{array}\right] \end{array}.$$

In the first case $D = 1/23$, in the second case $D = 5/23$. This shows the advantage of taking into account weights of bi-cliques. If we take the sequential automaton equivalent to the parallel automaton from Example 1 and apply, for instance, the method for low power state assignment [14], then $D = 3/23$. It allows one to say about utility of the suggested method.

7. Conclusion

The suggested method for low power state assignment of a parallel automaton is intended for using in a computer aided logical design system. Comparison of the results of applying the described method with the results of the partial state assignment without taking into account the switching activity of memory elements shows that the described method gives the better result. It is confirmed in considering 6 parallel automata with the numbers of partial states from 8 to 13, input variables from 2 to 3, and transitions 5, 6, 8, 9 and 13. Moreover, the Boolean function systems obtained after applying the described method for the state assignment turn out to be not worse than ones obtained without taking into account the switching activity of memory elements. This comparison is made in terms of the number of elementary conjunctions in disjunctive normal forms of the functions and the sum of their ranks.

REFERENCES

1. *Muroga S.* VLSI System Design. When and How to Design Very-Large-Scale Integrated Circuits. N.Y., John Wiley & Sons, 1982.
2. *Pedram M.* Power minimization in IC design: Principles and applications. ACM Trans. Design Automat. Electron. Syst., 1996, vol. 6, pp. 3–56.
3. *Kashirova L., Keevallik A., and Meshkov M.* State assignment of finite state machine for decrease of power dissipation. Second Intern. Conf. CAD DD'97, Minsk, Republic of Belarus, November 12–14, 1997. Minsk, NASB, Institute of Engineering Cybernetics, 1997, vol. 1, pp. 60–67.

4. *Sudnitson A.* Partition search for FSM low power synthesis. Fourth Intern. Conf. CAD DD'2001, Minsk, November 14–16, 2001. Minsk, NASB, Institute of Engineering Cybernetics, 2001, vol. 1, pp. 44–49.
5. *Zakrevskiy A. D.* Algoritmy energosberegayushchego kodirovaniya sostoyaniy avtomata [Algorithms for low power state assignment of an automaton]. Informatika, 2011, no. 1(29), pp. 68–78. (in Russian)
6. *Pottosin Yu. V.* Kodirovanie sostoyaniy diskretnogo avtomata, orientirovannoe na umen'shenie energopotrebleniya realizuyushchey skhemy [State assignment in a discrete automaton targeting an implementing low power circuit]. Prikladnaya Diskretnaya Matematika, 2011, no. 4(14), pp. 62–71. (in Russian)
7. *Pottosin Yu. V.* Energosberegayushchee protivogonochnoe kodirovanie sostoyaniy asinkhronnogo avtomata [Low power race-free state assignment of an asynchronous automaton]. Informatika, 2015, no. 2(46), pp. 94–101. (in Russian)
8. *Pottosin Yu.* Race-free state assignment for low power asynchronous automaton. Further Improvements in the Boolean Domain. Ed. B. Steinbach. Cambridge Scholars Publ., 2018, pp. 253–267.
9. *Pottosin Yu.* Optimal state assignment of synchronous parallel automata. Design of Embedded Control Systems. Eds. M. A. Adamski, A. Karatkevich, and M. Wegrzyn. N.Y., Springer, 2005, pp. 111–124.
10. *Zakrevskiy A. D.* Parallel'nye Algoritmy Logicheskogo Upravleniya [Parallel Algorithms for Logical Control]. Moscow, URSS, 2003, 304 p. (in Russian)
11. *Peterson J. L.* Petri Net Theory and the Modeling of Systems. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.
12. *Zakrevskiy A. D., Pottosin Yu. V., and Cheremisinova L. D.* Design of Logical Control Devices. Tallinn, TUT Press, 2009.
13. *Zakrevskiy A. D., Pottosin Yu. V., and Cheremisinova L. D.* Combinatorial Algorithms of Discrete Mathematics. Tallinn, TUT Press, 2008.
14. *Pottosin Yu. V.* Kombinatornye Zadachi v Logicheskom Proektirovanii Diskretnykh Ustroystv [Combinatorial Problems in Logical Design of Discrete Devices]. Minsk, Belaruskaja Navuka, 2021, 175 p. (in Russian)
15. *Zakrevskiy A. D.* Optimizatsiya pokrytiy mnozhestv [Optimization of set covering]. Logicheskiy Yazyk dlya Predstavleniya Algoritmov Sinteza Releynykh Ustroystv [Logical Language for Representation of Algorithms for Relay Devices Synthesis]. Ed. M. A. Gavrilov. Moscow, Nauka, 1966, pp. 136–148. (in Russian)

СВЕДЕНИЯ ОБ АВТОРАХ

ГАМАЮНОВ Денис Юрьевич — кандидат физико-математических наук, доцент факультета ВМК МГУ имени М. В. Ломоносова, г. Москва.
E-mail: **gamajun@seclab.cs.msu.su**

МАЛАХОВ Станислав Сергеевич — аспирант кафедры компьютерной безопасности Национального исследовательского университета «Высшая школа экономики», г. Москва. E-mail: **ssmalakhov@edu.hse.ru**

МАЛЫГИНА Екатерина Сергеевна — кандидат физико-математических наук, доцент ИФМНиИТ, младший научный сотрудник лаборатории «Математические методы защиты и обработки информации» научно-образовательного математического центра «Северо-Западный центр математических исследований имени Софьи Ковалевской» БФУ им. И. Канта, г. Калининград. E-mail: **emalygina@kantiana.ru**

НЕСТЕРЕНКО Алексей Юрьевич — кандидат физико-математических наук, доцент кафедры компьютерной безопасности Московского института электроники и математики им. А. Н. Тихонова Национального исследовательского университета «Высшая школа экономики» (МИЭМ НИУ ВШЭ), г. Москва.
E-mail: **anesterenko@hse.ru**

ПОПКОВ Кирилл Андреевич — кандидат физико-математических наук, научный сотрудник Института прикладной математики им. М. В. Келдыша РАН, г. Москва.
E-mail: **kirill-formulist@mail.ru**

ПОТТОСИН Юрий Васильевич — кандидат физико-математических наук, доцент, ведущий научный сотрудник Объединенного института проблем информатики Национальной академии наук Беларуси, г. Минск. E-mail: **pott@newman.bas-net.by**

РОЖКОВ Михаил Иванович — доктор технических наук, профессор кафедры компьютерной безопасности Национального исследовательского университета «Высшая школа экономики», г. Москва. E-mail: **mirozhkov@hse.ru**

РОМАНЬКОВ Виталий Анатольевич — доктор физико-математических наук, профессор, заведующий кафедрой Омского государственного университета им. Ф. М. Достоевского, г. Омск, профессор Сибирского федерального университета, г. Красноярск. E-mail: **romankov48@mail.ru**

СЕМЕНОВ Александр Михайлович — стажер-исследователь кафедры компьютерной безопасности Московского института электроники и математики им. А. Н. Тихонова Национального исследовательского университета «Высшая школа экономики» (МИЭМ НИУ ВШЭ), г. Москва. E-mail: **amsemenov@hse.ru**

ШЛЯХТИНА Елена Анатольевна — студентка факультета ВМК МГУ имени М. В. Ломоносова, программист ООО «Солидсофт», г. Москва.
E-mail: **elena.shlyakhtina@solidwall.io**

Журнал «Прикладная дискретная математика» входит в перечень ВАК рецензируемых научных изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание учёной степени кандидата и доктора наук по специальностям 01.01.06 — «Математическая логика, алгебра и теория чисел», 01.01.09 — «Дискретная математика и математическая кибернетика», 05.13.11 — «Математическое и программное обеспечение вычислительных машин,комплексов и компьютерных сетей», 05.13.17 — «Теоретические основы информатики», 05.13.19 — «Методы и системы защиты информации, информационная безопасность», а также в перечень журналов, рекомендованных ФУМО ВО ИБ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал индексируется в базах данных Web of Science (Emerging Sources Citation Index (ESCI) и Russian Science Citation Index (RSCI)), Scopus, MathSciNet и Zentralblatt MATH.

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте journals.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также правила подготовки рукописей статей для публикации в журнале.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надёжности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и её приложениям*