МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2017 № 35

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: vestnik_pdm@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 15.03.2017. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 13. Уч.-изд. л. 14,5.
Тираж 300 экз. Заказ № 2432. Цена свободная. Дата выхода в свет 1.04.2017.

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

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 512.542

ГОМОМОРФНАЯ УСТОЙЧИВОСТЬ КОНЕЧНЫХ ГРУПП

М. И. Кабенюк

Кемеровский государственный университет, г. Кемерово, Россия

Множество $\mathrm{Hom}\,(G,H)$ гомоморфизмов группы G в группу H является группой относительно поточечного умножения тогда и только тогда, когда образы любых двух таких гомоморфизмов поэлементно перестановочны. В таком случае группа $\mathrm{Hom}\,(G,H)$ коммутативна. Для конечных групп G и H изучаются алгебраические свойства группы $\mathrm{Hom}\,(G,H)$, а также объединения $\mathrm{Im}\,(G,H)$ образов всех таких гомоморфизмов. Пусть $\exp(G)$ — минимальное среди всех таких положительных целых чисел n, для которых $x^n = 1$ для каждого элемента $x \in G$; G' — коммутант группы G, $q = \exp(G/G')$ и $\Omega_q(H)$ — подгруппа в H, порождённая элементами периода q. Доказаны следующие утверждения:
– Если $\mathrm{Hom}\,(G,H)$ является группой, то $\Omega_q(H)$ коммутативна и группы $\mathrm{Hom}\,(G,H)$ и $\mathrm{Hom}\,(G/G',\Omega_q(H))$ изоморфны. Обратно, если $\Omega_q(H)$ коммутативна и ядро каждого гомоморфизма из $\mathrm{Hom}\,(G,H)$ содержит коммутант G', то множество $\mathrm{Hom}\,(G,H)$ является группой относительно поточечного умножения.
– Если $\mathrm{Im}\,(G,H)$ — подгруппа в H, то $\mathrm{Im}\,(G,H)$ эндоморфно допустима.
– Если G — такая конечная p-группа, что $q = \exp(G) = \exp(G/G')$, а H — регулярная p-группа, то $\mathrm{Im}\,(G,H) = \Omega_q(H)$.

Ключевые слова: *гомоморфизм групп, гомоморфная устойчивость, конечная группа, группа Фробениуса, регулярная p-группа.*

DOI 10.17223/20710410/35/1

HOMOMORPHIC STABILITY OF FINITE GROUPS

M. I. Kabenyuk

Kemerovo State University, Kemerovo, Russia

E-mail: kabenyuk@kemsu.ru

The set $\mathrm{Hom}\,(G,H)$ of all homomorphisms from a group G to a group H is a group with respect to the operation of pointwise products iff the images of any two such homomorphisms commute element-wise; in this case, the group is commutative. For finite G and H, we study algebraic properties of this group and of the union $\mathrm{Im}\,(G,H)$ of the images of all homomorphisms from G to H. Let $\exp(G)$ be the minimal positive integer n such that $x^n = 1$ for all $x \in G$, let G' be the commutator subgroup of G, $q = \exp(G/G')$, and let $\Omega_q(H)$ be the subgroup of H generated by all elements of order q. We obtain the following results.
If $\mathrm{Hom}\,(G,H)$ is a group, then $\Omega_q(H)$ is commutative and the groups $\mathrm{Hom}\,(G,H)$ and $\mathrm{Hom}\,(G/G',\Omega_q(H))$ are isomorphic. Conversely, if $\Omega_q(H)$ is commutative and $\phi(G') = \{1\}$ for all $\phi \in \mathrm{Hom}\,(G,H)$, then $\mathrm{Hom}\,(G,H)$ is a group.

If $\mathrm{Im}\,(G,H)$ is a subgroup of H, then it is endomorphically admissible in H.
If G is a finite p-group such that $\exp(G)=\exp(G/G')=q$ and H is a regular p-group, then $\mathrm{Im}\,(G,H)=\Omega_q(H)$.

Keywords: *homomorphism groups, homomorphic stability, finite group, Frobenius group, regular p-group.*

Введение

Пусть G и H — произвольные группы. Если ϕ и ψ — произвольные отображения, заданные на G и принимающие значения в H, то их произведением назовём отображение $\phi\psi:\ G\to H$, определяемое правилом

$$(\phi\psi)(x)=\phi(x)\psi(x),\ \ x\in G. \tag{1}$$

Символом $\mathrm{Hom}\,(G,H)$ будем обозначать множество всех гомоморфизмов группы G в группу H. Если H — абелева группа, то множество $\mathrm{Hom}\,(G,H)$ является группой относительно указанного выше умножения отображений. Эта группа тоже коммутативна. Если же H не является абелевой группой, то $\mathrm{Hom}\,(G,H)$ не обязано быть группой. Пару групп (G,H) назовём гомоморфно устойчивой, если объединение гомоморфных образов группы G в группе H является подгруппой группы H, т. е. множество

$$\mathrm{Im}\,(G,H)=\bigcup_{\phi\in\mathrm{Hom}\,(G,H)}\mathrm{Im}\,\phi$$

— подгруппа группы H. Это понятие рассматривается в работе [1]. Там же, а также в [2] исследуется вопрос о гомоморфной устойчивости пар абелевых групп. В частности, в [2, следствие 2] доказано, что пара (G,H) абелевых групп гомоморфно устойчива, если группа G периодическая, а H — произвольная группа (этот факт упоминается в [3, гл. VIII, § 43, упражнение 11]).

В [4] понятие гомоморфной устойчивости рассматривается для произвольных пар конечных групп. В [4, 5] для всех пар (G,H), $|G|\leqslant 12$, $|H|\leqslant 12$, с помощью компьютерных программ установлено, какие из них являются гомоморфно устойчивыми; в том случае, когда это так, вычислена группа $\mathrm{Im}\,(G,H)$. Кроме того, для всех указанных пар установлено, какие из множеств $\mathrm{Hom}\,(G,H)$ являются группами, и все эти группы вычислены.

В настоящей работе доказано несколько общих утверждений о множествах $\mathrm{Hom}\,(G,H)$ и $\mathrm{Im}\,(G,H)$, которые позволяют проделать все вычисления указанных выше работ без привлечения компьютерных программ и исправить несколько опечаток, допущенных авторами. Помимо этого, эти утверждения могли бы помочь проделать указанные вычисления для групп более высокого порядка.

Все используемые обозначения общеприняты. Редкие исключения разъяснены в тексте. Для группы G и $x\in G$ символы $|G|$ и $\mathrm{o}(x)$ обозначают порядок G и период (часто говорят «порядок») элемента x. Пусть G — конечная группа. *Высотой* (используют также термин «экспонента») группы G назовём такое целое положительное число q, что $x^q=1$ для любого элемента $x\in G$, и число q минимально с этим свойством. Высоту группы G будем обозначать символом $\exp(G)$. В частности, высота группы G — делитель $|G|$. Если m, n — целые числа, то символ $m|n$ означает, что m — делитель n. Для конечных абелевых групп и конечных p-групп A верно следующее равенство:

$$\exp(A)=\max_{x\in A}\{\mathrm{o}(x)\}.$$

С другой стороны, $\exp(S_3) = 6$, но в группе S_3 нет элементов периода 6; $\exp(A_5) = 30$, но максимальный период элементов A_5 равен 5. Пусть q — некоторое целое положительное число, H — некоторая группа. Будем обозначать

$$\Omega_q(H) = \mathrm{gr}\,(x \in H \mid x^q = 1).$$

Здесь и далее символом $\mathrm{gr}\,(X)$ обозначается подгруппа, порождённая множеством $X \subset H$. Символ 1 обозначает как нейтральный элемент мультипликативной группы, так и группу, состоящую из одного нейтрального элемента.

1. Множество гомоморфизмов пар конечных групп

Теорема 1. Пусть G и H — произвольные группы. Множество $\mathrm{Hom}\,(G, H)$ является группой относительно умножения (1) тогда и только тогда, когда образы $\mathrm{Im}\,\phi$ и $\mathrm{Im}\,\psi$ поэлементно перестановочны для любых $\phi, \psi \in \mathrm{Hom}\,(G, H)$.

Доказательство. Пусть множество $\mathrm{Hom}\,(G, H)$ является группой относительно умножения, определяемого формулой (1). Это означает, что если $\phi, \psi \in \mathrm{Hom}\,(G, H)$, то их произведение $\theta = \phi\psi$ — тоже гомоморфизм. Следовательно,

$$\theta(xy) = \theta(x)\theta(y) \tag{2}$$

для любых $x, y \in G$. Учитывая формулу (1) и тот факт, что ϕ и ψ — гомоморфизмы, из (2) получаем

$$\phi(x)\phi(y)\psi(x)\psi(y) = \phi(x)\psi(x)\phi(y)\psi(y). \tag{3}$$

Сократив (3) на $\phi(x)$ слева и на $\psi(y)$ справа, получим

$$\phi(y)\psi(x) = \psi(x)\phi(y) \tag{4}$$

для любых $x, y \in G$ и любых $\phi, \psi \in \mathrm{Hom}\,(G, H)$. Этим доказано прямое утверждение теоремы 1.

Для доказательства обратного утверждения последовательно переходим от (4) к (3), а затем к (2). В результате убеждаемся, что произведение $\theta = \phi\psi$ является гомоморфизмом группы G в группу H, если ϕ и ψ были такими. ■

Теорема 2. Если множество $\mathrm{Hom}\,(G, H)$ является группой, то
(i) образ $\mathrm{Im}\,\phi$ — коммутативная подгруппа в H для любого $\phi \in \mathrm{Hom}\,(G, H)$;
(ii) группа $\mathrm{Hom}\,(G, H)$ абелева.

Доказательство. Пункт (i) этой теоремы вытекает из формулы (4), где следует положить $\psi = \phi$.

Для доказательства п. (ii) вычислим по формуле (1) выражения $(\phi\psi)(x)$ и $(\psi\phi)(x)$, а затем из формулы (4) получим $(\phi\psi)(x) = (\psi\phi)(x)$ для любого $x \in G$. ■

Замечание 1. Последнее утверждение показывает, что в таблице 3 из работы [4] (таблица 5 из [5]) имеются неточности, требующие исправления для следующих пяти множеств: $\mathrm{Hom}\,(\mathbb{Z}_6, D_3)$, $\mathrm{Hom}\,(\mathbb{Z}_6, D_6)$, $\mathrm{Hom}\,(\mathbb{Z}_{12}, \mathbb{Q}_8)$, $\mathrm{Hom}\,(T_{12}, D_4)$, $\mathrm{Hom}\,(T_{12}, \mathbb{Q}_8)$. Здесь $\mathbb{Z}_n$ — циклическая группа порядка n; D_n — диэдральная группа порядка $2n$; $\mathbb{Q}_8$ — группа кватернионов порядка 8; T_{12} — группа, задаваемая с помощью порождающих и соотношений следующим образом:

$$T_{12} = \mathrm{gr}\,(a, b \mid a^3 = b^4 = 1,\ ba = ab^{-1}).$$

Ниже мы укажем, каким образом должны быть исправлены эти неточности.

Далее потребуется одно утверждение об абелевых группах.

Лемма 1. Пусть B — конечная абелева группа, $q = \exp(B)$. Тогда для любого целого числа n, для которого $n|q$, существует такая подгруппа $C < B$, что факторгруппа B/C — циклическая группа порядка n.

Доказательство. Известно, что каждая конечная абелева группа является прямым произведением конечного числа циклических групп, имеющих порядки, равные степеням простых чисел [3, теорема 15.2, гл. III, § 15]. Если $q = p_1^{k_1} \dots p_s^{k_s}$, где $p_1, \dots, p_s$ — попарно различные простые числа, то это прямое произведение должно иметь вид

$$B = B_1 \times \dots \times B_s \times D,$$

где $B_1, \dots, B_s$ — циклические группы порядков $p_1^{k_1}, \dots, p_s^{k_s}$ соответственно, а D — некоторая подгруппа B. Группа $X = B_1 \times \dots \times B_s$ является циклической группой порядка $q = p_1^{k_1} \dots p_s^{k_s}$. Итак, $B = X \times D$, $X = \operatorname{gr}(x)$ для некоторого $x \in X$, $\mathrm{o}(x) = q$. Искомая подгруппа — $C = \operatorname{gr}(x^m) \times D$, где $m = q/n$. ■

Теорема 3. Пусть G и H — две группы, G' — коммутант группы G, $q = \exp(G/G')$, $A = \Omega_q(H)$. Если множество $\operatorname{Hom}(G, H)$ является группой, то A — коммутативная подгруппа в H и

$$\operatorname{Hom}(G, H) \approx \operatorname{Hom}(G/G', A). \tag{5}$$

Обратно, если $\Omega_q(H)$ — коммутативная подгруппа и ядро каждого гомоморфизма из $\operatorname{Hom}(G, H)$ содержит коммутант G', то множество $\operatorname{Hom}(G, H)$ является группой относительно умножения, заданного формулой (1).

Доказательство. Обозначим $B = G/G'$. Пусть $x \in A$. Так как $\mathrm{o}(x)|q$, то по лемме 1 существует такая подгруппа $C < B$, что B/C — циклическая группа порядка $\mathrm{o}(x)$. Пусть $\phi : G \longrightarrow H$ — гомоморфизм, являющийся композицией

$$G \xrightarrow{\varepsilon_1} G/G' = B \xrightarrow{\varepsilon_2} B/C \xrightarrow{i} \operatorname{gr}(x) < H.$$

Здесь $\varepsilon_1, \varepsilon_2$ — канонические гомоморфизмы; i — изоморфизм двух циклических групп одного порядка $\mathrm{o}(x)$. Видим, что $x \in \operatorname{Im}\phi$. Аналогично может быть построен гомоморфизм $\psi : G \longrightarrow H$, для которого $y \in \operatorname{Im}\psi$. По теореме 1 множества $\operatorname{Im}\phi$ и $\operatorname{Im}\psi$ поэлементно перестановочны. В частности, $xy = yx$. Этим доказано, что $A = \Omega_q(H)$ — коммутативная подгруппа в H.

Докажем формулу (5). Пусть $\phi \in \operatorname{Hom}(G, H)$. Так как $\operatorname{Hom}(G, H)$ является по условию группой, по теореме 2, (i) $\operatorname{Im}\phi$ — коммутативная подгруппа в H. Следовательно, $G' \subset \operatorname{Im}\phi$. Определим отображение $\widetilde{\phi} : G/G' \longrightarrow H$ по формуле

$$\widetilde{\phi}(\widetilde{x}) = \phi(x),$$

где $x \in G$, $\widetilde{x} = xG'$. Корректность определения $\widetilde{\phi}$ следует из включения $G' \subset \operatorname{Im}\phi$. Непосредственно проверяется, что $\widetilde{\phi}$ — гомоморфизм группы G/G' в группу H. Так как $n = \mathrm{o}(\phi(x)) = \mathrm{o}(\widetilde{\phi}(\widetilde{x}))$ и $\widetilde{x} \in G/G'$, то $n|\exp(G/G')$. Стало быть, $\phi(x) \in A = \Omega_q(H)$. Последнее верно для любого $x \in G$. Следовательно, $\widetilde{\phi} \in \operatorname{Hom}(G/G', A)$, а отображение $\phi \mapsto \widetilde{\phi}$ является изоморфизмом группы $\operatorname{Hom}(G, H)$ на группу $\operatorname{Hom}(G/G', A)$.

Докажем обратное утверждение. Пусть множество $A = \Omega_q(H)$ является подгруппой в H. (Между прочим, отметим, хотя это нам и не потребуется, что тогда A — нормальная подгруппа в H.) Достаточно доказать, что $\operatorname{Im}\phi \subset A$ для любого $\phi \in \operatorname{Hom}(G, H)$.

Так как ядро ϕ содержит G', то $\operatorname{Im}\phi = \phi(G)$ — абелева группа. В силу определения высоты q для любого элемента $x \in G$ выполняется условие $\mathrm{o}(x)|q$. Так как $\mathrm{o}(\phi(x))|\mathrm{o}(x)$, то $\mathrm{o}(\phi(x))|q$. Значит, $\phi(x) \in A$. ■

Замечание 2. Теорема 3 часто помогает свести вычисление группы $\operatorname{Hom}(G,H)$ к точно такой же задаче, но уже для абелевых групп. Последнее позволяет указать следующий алгоритм вычисления группы $\operatorname{Hom}(G,H)$:

1) Вычисляем коммутант G' группы G.
2) Вычисляем высоту $q = \exp(G/G')$ фактор-группы G/G'.
3) Вычисляем множество $X = \{x \in H : x^q = 1\}$.
4) Если элементы X попарно перестановочны, то переходим к шагу 5, иначе заканчиваем вычисления, так как в этом случае множество $\operatorname{Hom}(G,H)$ не является группой.
5) Проверяем наличие гомоморфизма $\phi : G \to H$, для которого $G' \not\subseteq \operatorname{Ker}\phi$. Если такого гомоморфизма нет, то переходим к шагу 6. Если такой гомоморфизм есть, то множество $\operatorname{Hom}(G,H)$ не является группой и вычисления заканчиваем.
6) Вычисляем $\operatorname{Hom}(G/G', A)$, где $A = \Omega_q(H)$. В силу формулы (5) это и есть $\operatorname{Hom}(G,H)$.

Отметим, что шаг 5 может оказаться довольно сложной задачей для групп большого порядка. Мы не обсуждаем здесь пути решения этой задачи.

В то же время шаг 6 может быть легко выполнен благодаря следующему обстоятельству. Для конечных абелевых групп B и C, порядки которых взаимно просты, $\operatorname{Hom}(B,C) = 0$. Если же обе группы B и C являются p-группами для некоторого простого числа p, то $\operatorname{Hom}(B,C)$ вычисляется по следующей формуле: пусть

$$B = \mathbb{Z}_{p^{k_1}} \oplus \mathbb{Z}_{p^{k_2}} \oplus \ldots \oplus \mathbb{Z}_{p^{k_r}},\ C = \mathbb{Z}_{p^{l_1}} \oplus \mathbb{Z}_{p^{l_2}} \oplus \ldots \oplus \mathbb{Z}_{p^{l_s}}.$$

Здесь $k_1, k_2, \ldots, k_r, l_1, l_2, \ldots, l_s$ — целые положительные числа. Тогда справедлива формула

$$\operatorname{Hom}(B,C) \approx \mathbb{Z}_{p^{m_{11}}} \oplus \mathbb{Z}_{p^{m_{12}}} \oplus \ldots \oplus \mathbb{Z}_{p^{m_{1s}}} \oplus \ldots \oplus \mathbb{Z}_{p^{m_{r1}}} \oplus \mathbb{Z}_{p^{m_{r2}}} \oplus \ldots \oplus \mathbb{Z}_{p^{m_{rs}}}, \quad (6)$$

где $m_{ij} = \min(k_i, l_j)$, $i = 1, \ldots, r$, $j = 1, \ldots, s$.

Формула (6) — простое следствие теорем 43.1 и 43.2 из [3, гл. VIII, § 43].

Следствие 1. Если G — конечная группа, H — конечная абелева группа, то $\operatorname{Hom}(G,H)$ — группа и

$$\operatorname{Hom}(G,H) \approx \operatorname{Hom}(G/G', H). \quad (7)$$

Доказательство. Тот факт, что $\operatorname{Hom}(G,H)$ — группа, немедленно следует из формулы умножения (1) на множестве $\operatorname{Hom}(G,H)$. Вторая часть утверждения — прямое следствие формулы (5) теоремы 3. ■

Следствие 2. Пусть G — конечная абелева группа, H — произвольная конечная группа, $q = \exp(G)$ — высота группы G, $A = \Omega_q(H)$. Тогда $\operatorname{Hom}(G,H)$ является группой тогда и только тогда, когда A — коммутативная подгруппа H. Если это условие выполняется, то

$$\operatorname{Hom}(G,H) \approx \operatorname{Hom}(G,A).$$

Доказательство. Следствие верно ввиду того, что для абелевых групп G коммутант $G' = 1$. ■

2. Приложения общих утверждений

Проиллюстрируем применение доказанных утверждений для вычисления конкретных групп $\mathrm{Hom}\,(G,H)$. Рассмотрим таблицу 1 из [4] (или [5, таблица 3]). В ней перечислены все группы $\mathrm{Hom}\,(G,H)$ для всех конечных групп G и абелевых групп H, для которых $|G|,|H|\leqslant 12$. Эта таблица составлена с помощью компьютерной программы. Укажем, как она может быть заполнена с применением наших утверждений.

Если группа G абелева, то соответствующие строки этой таблицы можно заполнить, используя формулу (6).

Если группа G неабелева, то воспользуемся формулой (7). Для этого придётся вычислить фактор-группы G/G' всех неабелевых групп, порядок которых не больше 12. Таких групп ровно семь:

$$D_3,\ D_4,\ D_5,\ D_6,\ \mathbb{Q}_8,\ A_4,\ T_{12}.$$

Первые четыре группы — диэдральные группы соответствующего порядка. Для диэдральных групп имеем следующие общие формулы:

$$D_{2n}/D'_{2n}\approx\mathbb{Z}_2\oplus\mathbb{Z}_2,\ \ D_{2n-1}/D'_{2n-1}\approx\mathbb{Z}_2,\ \ n=2,3,4,\ldots \tag{8}$$

Далее

$$\mathbb{Q}_8/\mathbb{Q}'_8\approx\mathbb{Z}_2\oplus\mathbb{Z}_2,\ \ A_4/A'_4\approx\mathbb{Z}_2,\ \ T_{12}/T'_{12}\approx\mathbb{Z}_4. \tag{9}$$

Поясним последний изоморфизм. Поскольку $T_{12}=\{a,b\mid a^3=b^4=1,\ ba=ab^{-1}\}$, то $a^{-1}ba=b^{-1}$ и $b^{-1}a^{-1}ba=b$. Из первого равенства следует, что подгруппа $\mathrm{gr}\,(b)$ нормальна в T_{12}, а из второго — $\mathrm{gr}\,(b)\subset T'_{12}$. Отсюда выводим, что $\mathrm{gr}\,(b)=T'_{12}$ и $T_{12}/T'_{12}\approx\mathbb{Z}_4$.

Из (7)–(9) получаем, что для любой абелевой группы H имеют место изоморфизмы

$$\begin{gathered}\mathrm{Hom}\,(D_3,H)\approx\mathrm{Hom}\,(D_5,H)\approx\mathrm{Hom}\,(A_4,H)\approx\mathrm{Hom}\,(\mathbb{Z}_2,H),\\ \mathrm{Hom}\,(D_4,H)\approx\mathrm{Hom}\,(D_6,H)\approx\mathrm{Hom}\,(\mathbb{Q}_8,H)\approx\mathrm{Hom}\,(\mathbb{Z}_2\oplus\mathbb{Z}_2,H),\\ \mathrm{Hom}\,(T_{12},H)\approx\mathrm{Hom}\,(\mathbb{Z}_4,H).\end{gathered}$$

Отсюда следует, что должны полностью совпадать строки таблицы 1 из [4] (или [5, таблица 3]), помеченные $\mathbb{Z}_2, D_3, D_5, A_4$, а также строки $\mathbb{Z}_2^2, D_4, D_6, \mathbb{Q}_8$ и, наконец, строки $\mathbb{Z}_4, T_{12}$. И это действительно так, за исключением одного места, находящегося в строке $\mathbb{Z}_2^2$ и столбце $\mathbb{Z}_2$, где вместо $\mathbb{Z}_4$ должно стоять $\mathbb{Z}_2^2$.

Проиллюстрируем, как может работать теорема 3 и её следствия в случае, когда группа H неабелева. Рассмотрим в качестве H те же семь подгрупп. Укажем в каждой из них нетривиальную коммутативную подгруппу, равную $\Omega_q(H)$ для некоторого целого положительного числа q.

В группе D_3 такая подгруппа только одна: $\Omega_3(D_3)\approx\mathbb{Z}_3$. В группе D_4 таких подгрупп нет. В D_5: $\Omega_5(D_5)\approx\mathbb{Z}_5$. В D_6: $\Omega_3(D_6)\approx\mathbb{Z}_3$. В $\mathbb{Q}_8$: $\Omega_2(\mathbb{Q}_8)\approx\mathbb{Z}_2$. В A_4: подгруппа $V_4=\Omega_2(A_4)\approx\mathbb{Z}_2^2$. В T_{12} таких подгрупп три: $\Omega_2(T_{12})\approx\mathbb{Z}_2$, $\Omega_3(T_{12})\approx\mathbb{Z}_3$, $\Omega_6(T_{12})\approx\mathbb{Z}_6$.

Используя эту информацию, благодаря теореме 3 и её следствиям, можем выписать группы $\mathrm{Hom}\,(G,H)$ или указать, что это множество не является группой. Рассмотрим только случаи, соответствующие клеткам таблицы 3 из работы [4], где, по нашему мнению, содержатся неточности. Итак,

$$\begin{gathered}\mathrm{Hom}\,(\mathbb{Z}_6,D_3)\approx\mathrm{Hom}\,(\mathbb{Z}_6,\mathbb{Z}_3)\approx\mathbb{Z}_3,\\ \mathrm{Hom}\,(\mathbb{Z}_6,D_6)\approx\mathrm{Hom}\,(\mathbb{Z}_6,\mathbb{Z}_3)\approx\mathbb{Z}_3,\\ \mathrm{Hom}\,(\mathbb{Z}_{12},\mathbb{Q}_8)\approx\mathrm{Hom}\,(\mathbb{Z}_{12},\mathbb{Z}_2)\approx\mathbb{Z}_2.\end{gathered}$$

Множество $\mathrm{Hom}\,(T_{12}, D_4)$ группой не является, поскольку в группе D_4 нет нетривиальных коммутативных подгрупп вида $\Omega_q(D_4)$. Отметим, однако, что $\mathrm{Hom}\,(T_{12}, D_4)$ содержит нетривиальные гомоморфизмы.

Вычислим $\mathrm{Hom}\,(T_{12}, \mathbb{Q}_8)$. Мы знаем, что

$$T_{12}/T'_{12} \approx \mathbb{Z}_4,\ A = \Omega_2(\mathbb{Q}_8) < \mathbb{Q}_8,\ A \approx \mathbb{Z}_2.$$

Ясно, что ядро каждого гомоморфизма из T_{12} в $\mathbb{Q}_8$ содержит коммутант. Поэтому $\mathrm{Hom}\,(T_{12}, \mathbb{Q}_8) \approx \mathrm{Hom}\,(\mathbb{Z}_4, \mathbb{Z}_2) \approx \mathbb{Z}_2$.

При этом в [4, таблица 3] на этих местах стоят соответственно D_3, D_3, $\mathbb{Q}_8$, D_4, $\mathbb{Q}_8$.

3. Гомоморфная устойчивость регулярных p-групп

Из приводимого ниже утверждения следует, в частности, что понятие гомоморфной устойчивости и понятие сильной гомоморфной устойчивости, введённое в [4], равносильны между собой.

Теорема 4. Если пара (G, H) гомоморфно устойчива, т. е. если $\mathrm{Im}\,(G, H)$ — подгруппа в H, то $\mathrm{Im}\,(G, H)$ — эндоморфно допустимая подгруппа группы H. В частности, $\mathrm{Im}\,(G, H)$ — нормальная подгруппа группы H.

Доказательство. Ясно, что если $\phi : G \to H$ и $\psi : H \to H$ — гомоморфизмы, то композиция $\theta = \psi \circ \phi$ — гомоморфизм группы G в группу H. Пусть $x \in \mathrm{Im}\,(G, H)$. Тогда существуют такие $\phi \in \mathrm{Hom}\,(G, H)$ и $y \in G$, что $\phi(y) = x$. Значит, для любого гомоморфизма $\psi : H \to H$ имеем

$$(\psi \circ \phi)(y) = \psi(\phi(y)) = \psi(x),$$

то есть $\psi(x) \in \mathrm{Im}\,(\psi \circ \phi) \subset \mathrm{Im}\,(G, H)$. Этим доказано, что вместе с каждым элементом в $\mathrm{Im}\,(G, H)$ лежат все его гомоморфные образы. Стало быть, если $\mathrm{Im}\,(G, H)$ — подгруппа группы H, то она эндоморфно допустима. ■

Далее приведём несколько примеров и докажем несколько общих утверждений о множестве $\mathrm{Im}\,(G, H)$.

Утверждение 1. Если $n = \exp(H)$, то пара $(\mathbb{Z}_n, H)$ гомоморфно устойчива. Более точно, $\mathrm{Im}\,(\mathbb{Z}_n, H) = H$.

Доказательство. Пусть $h \in H$ и $\mathrm{o}(h) = m$. Так как по условию $h^n = 1$, то $m|n$. Тогда формула $\phi(k) = h^k$, $k \in \mathbb{Z}_n$, корректно определяет гомоморфизм из группы $\mathbb{Z}_n$ в группу H, причём $\phi(1) = h$. Значит, $h \in \mathrm{Im}\,(\mathbb{Z}_n, H)$. Ввиду произвольности h утверждение доказано. ■

Например, $\mathrm{Im}\,(\mathbb{Z}_{30}, A_5) = A_5$. Здесь и далее A_n — группа чётных перестановок, S_n — группа всех перестановок порядка n, V_4 — четверная группа Клейна. Приведем ещё несколько примеров, опуская вычисления.

Пример 1.

1) $\mathrm{Im}\,(S_3, A_4) = V_4$;
2) $\mathrm{Im}\,(A_4, S_3) = \mathrm{gr}\,(123)$;
3) $\mathrm{Im}\,(S_3, S_4) = \{x \in S_4 | \mathrm{o}(x) \neq 4\}$;
4) $\mathrm{Im}\,(S_4, S_3) = S_3$;
5) $\mathrm{Im}\,(S_4, A_4) = V_4$;
6) $\mathrm{Im}\,(A_4, S_4) = A_4$;
7) $\mathrm{Im}\,(A_5, A_5 \times A_5) = \{(x, y) | \mathrm{o}(x) = \mathrm{o}(y)\}$.

Таким образом, среди перечисленных пар только две — (S_3, S_4) и $(A_5, A_5 \times A_5)$ — не являются гомоморфно устойчивыми.

Укажем ещё один важный, на наш взгляд, пример.

Пример 2. Пусть $H = A \cdot B$ — группа Фробениуса с ядром A и дополнением B, $q = \exp(A)$. Тогда $\operatorname{Im}(\mathbb{Z}_q \times B, H) = H$, то есть пара $(\mathbb{Z}_q \times B, H)$ гомоморфно устойчива.

Подробные определения и обширные списки примеров групп Фробениуса можно найти в [6, 7]. Для объяснения нашего примера важно лишь следующее свойство группы Фробениуса $H = A \cdot B$:

$$H \setminus A \subset \bigcup_{x \in H} xHx^{-1}.$$

На наш взгляд, интересна следующая задача.

Задача 1. Для заданной конечной группы H вычислить все такие группы G, $|G| \leqslant |H|$, что пара (G, H) гомоморфно устойчива.

Другой пример гомоморфно устойчивых пар групп доставляют регулярные p-группы. Для подробного ознакомления с этим понятием, а также с многочисленными примерами p-групп можно обратиться к [8, глава 12]. Укажем лишь самое необходимое. Конечная p-группа называется регулярной, если для любых двух элементов $x, y \in G$ существует такой элемент $z \in H'$, где $H = \operatorname{gr}(x, y)$, что

$$x^p \cdot y^p = (xy)^p z^p.$$

Вот три семейства регулярных p-групп: абелевы p-группы; p-группы, ступень нильпотентности которых меньше p; p-группы, порядок которых меньше p^p.

Нам потребуется следующее свойство регулярных p-групп: если G — регулярная p-группа, то для любого целого числа вида $q = p^k$ группа $A = \Omega_q(G)$ является нормальной подгруппой, для которой $q = \exp(A)$ [8, теорема 12.4.5].

Теорема 5. Пусть G и H — такие конечные p-группы, что $\exp(G) = \exp(G/G')$, а H — регулярная p-группа. Тогда пара (G, H) гомоморфно устойчива. Более точно:

$$\operatorname{Im}(G, H) = \Omega_q(H), \ q = \exp(G).$$

Доказательство. Пусть $h \in H$ и $n = \mathrm{o}(h)$. Если $h \in \Omega_q(H)$, то $h^q = 1$. Тогда $n|q$. Так как $q = \exp(G/G')$, по лемме 1 существует такая подгруппа $C < B = G/G'$, что фактор-группа B/C — циклическая группа порядка n. Далее, рассуждая, как в доказательстве теоремы 3, построим гомоморфизм $\phi : G \to H$, такой, что $h \in \operatorname{Im}\phi$. Этим доказано, что $\operatorname{Im}(G, H) = \Omega_q(H)$. ■

В заключение сформулируем еще одну задачу.

Задача 2. Можно ли в теореме 5 ослабить условие $\exp(G) = \exp(G/G')$? Например, вообще убрать его?

ЛИТЕРАТУРА

1. *Гриншпон С. Я., Ельцова Т. А.* Гомоморфно устойчивые абелевы группы // Вестник Томского государственного университетата. 2003. № 280. С. 31–33.
2. *Гриншпон С. Я., Ельцова Т. А.* Гомоморфная устойчивость абелевых групп // Фундаментальная и прикладная математика. 2008. Т. 14. № 5. С. 67–76.
3. *Фукс Л.* Бесконечные абелевы группы. Т. 1. М.: Мир, 1974. 336 с.
4. *Шилин И. А., Китюков В. В.* Гомоморфная устойчивость пар групп малого порядка // Прикладная дискретная математика. 2011. № 4(14). С. 22–27.

5. *Шилин И. А., Китюков В. В., Александров А. А.* Вычисление групп гомоморфизмов и проверка гомоморфной устойчивости пар конечных групп // Прикладная информатика. 2012. Т. 37. № 1. С. 111–115.
6. *Brown R.* Frobenius Groups and Classical Maximal Orders. Amer. Math. Soc. 2001. 110 p.
7. *Perumal P.* On the Theory of the Frobenius Groups. `http://researchspace.ukzn.ac.za/handle/10413/8853`, 2012
8. *Холл М.* Теория групп. М.: ИЛ, 1962. 468 с.

REFERENCES

1. *Grinshpon S. Ya. and Yeltsova T. A.* Gomomorfno ustoychivye abelevy gruppy [Homomorphly stable Abelian groups]. Tomsk State University Journal, 2003, no. 280, pp. 31–33. (in Russian)
2. *Grinshpon S. Ya. and Yeltsova T. A.* Gomomorfnaya ustoychivost' abelevykh grupp [Homomorphic images of Abelian groups]. Fundam. Prikl. Mat., 2008, vol. 14, iss. 5, pp. 67–76. (in Russian)
3. *Fuchs L.* Infinite Abelian Groups, vol. 1. N. Y., San Francisco, London, Academic Press, 1970, 289 p.
4. *Shilin I. A. and Kityukov V. V.* Gomomorfnaya ustoychivost' par grupp malogo poryadka [Homomorphic stability of pairs of small order groups]. Prikladnaya Diskretnaya Matematika, 2011, no. 4(14), pp. 22–27.
5. *Shilin I. A., Kityukov V. V., and Aleksandrov A. A.* Vychislenie grupp gomomorfizmov i proverka gomomorfnoy ustoychivosti par konechnykh grupp [Homomorphism groups computing and homomorphic stability of pairs of finite groups verification]. Prikladnaya Informatika, 2012, vol. 37, no. 1, pp. 111–115. (in Russian)
6. *Brown R.* Frobenius Groups and Classical Maximal Orders. Amer. Math. Soc., 2001, 110 p.
7. *Perumal P.* On the Theory of the Frobenius Groups. `http://researchspace.ukzn.ac.za/handle/10413/8853`, 2012
8. *Hall M.* The Theory of Groups. Macmillan, 1959, 434 p.

УДК 519.214.5

ОЦЕНКА ДЛЯ РАСПРЕДЕЛЕНИЯ ЧИСЕЛ СЕРИЙ В СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ, УПРАВЛЯЕМОЙ СТАЦИОНАРНОЙ ЦЕПЬЮ МАРКОВА

Н. М. Меженная

Московский государственный технический университет имени Н. Э. Баумана, г. Москва, Россия

Проведено исследование асимптотических свойств совместного распределения чисел серий из разных знаков в последовательности случайных величин с полиномиальными распределениями, управляемой стационарной цепью Маркова с конечным числом состояний. Получена оценка расстояния по вариации между распределением случайного вектора из чисел серий заданных знаков и заданной длины в управляемой последовательности и сопровождающим многомерным распределением Пуассона. При доказательстве использованы метод Чена — Стейна, а также оценки расстояния по вариации между смешанным и обычным распределениями Пуассона. Из полученной оценки расстояния по вариации выведены многомерная пуассоновская и нормальная предельные теоремы для указанного случайного вектора.

Ключевые слова: *число серий, цепь Маркова, расстояние по вариации, метод Чена — Стейна, смешанное распределение Пуассона, предельная теорема Пуассона, центральная предельная теорема, скрытая марковская модель.*

DOI 10.17223/20710410/35/2

ESTIMATOR FOR THE DISTRIBUTION OF THE NUMBERS OF RUNS IN A RANDOM SEQUENCE CONTROLLED BY STATIONARY MARKOV CHAIN

N. M. Mezhennaya

Bauman Moscow State Technical University

E-mail: natalia.mezhennaya@gmail.com

The sequences of random characters from a finite set $\mathcal{A}$ with polynomial distributions controlled by a stationary finite-state Markov chain are considered. For numbers of character runs in them, the asymptotic properties of joint distributions are studied. We deduce an estimate for the total variation distance ρ_{TV} between the distribution of a random vector $\varsigma_{\mathcal{A}}$ with components being numbers of runs in a controlled sequence of an enough length T and accompanying multidimensional Poisson distribution $\mathrm{Pois}(\lambda_{\mathcal{A}})$. The estimate is $\rho_{TV}\left(\mathcal{L}(\varsigma_{\mathcal{A}}), \mathrm{Pois}(\lambda_{\mathcal{A}})\right) \leqslant \gamma\left(\gamma T (p^*)^{s_*} + 1\right)$, where $\gamma^2 = |\mathcal{A}|^2(2s^* + 3)(p^*)^{s_*}$, s_* (s^*) is the minimum (maximum) length of run in the set of components of the vector $\varsigma_{\mathcal{A}}$, and p^* is the maximum character probability in distributions given on $\mathcal{A}$. For deriving this estimate, we use the functional variant of Chen — Stein method and an estimation for the total variation distance between the mixed and ordinal Poisson distributions. This estimation is a function of the variance of mixing parameter of mixed Poisson distribution. Using the derived estimate for

the total variation distance ρ_{TV}, we deduce the multidimensional Poisson and normal limit theorems for the random vector $\varsigma_{\mathcal{A}}$ under appropriate conditions for scheme parameters.

Keywords: *number of runs, Markov chain, total variation distance, Chen — Stein method, mixed Poisson distribution, Poisson limit theorem, normal limit theorem, hidden Markov model.*

Введение

Статистические свойства чисел серий в дискретных случайных последовательностях широко используются в задачах передачи информации, контроля качества, анализа генома и т. д. (см. [1, глава 1] и библиографию там же). Вероятностные свойства связанных с ними статистик (чисел серий, длин промежутков между сериями и т. п.) в последовательностях, образованных независимыми или слабо зависимыми случайными величинами, хорошо изучены [1, главы 5, 7; 2]. Имеются как точные, так и асимптотические результаты.

В ряде работ изучены свойства чисел серий и цепочек специального вида из фиксированных знаков в марковских последовательностях. Например, в [3, 4] исследованы распределения времени до появления серии в цепи Маркова с двумя состояниями, а также распределения ряда других случайных величин, связанных с моментами появления серий в такой последовательности. В работах [5 – 7] найдено распределение чисел серий в марковской случайной последовательности с двумя и тремя состояниям. В [8] получены точные и приближённые формулы для вероятностей появления чисел серий из единиц в отрезках марковских случайных последовательностей с двумя состояниями и выражения для их математического ожидания и дисперсии. В [8, 9] подробно описано использование статистик, связанных с числом серий, для анализа последовательностей из зависимых случайных величин. В [10] получена производящая функция совместных вероятностей появления серий в цепи Маркова с конечным множеством состояний. Она позволяет найти явные выражения для вероятностей и числовых характеристик.

В работах [11 – 13] рассмотрен вопрос об оценках скорости сходимости для распределения чисел серий и цепочек в марковской цепи к сложному пуассоновскому распределению. Предельное поведение наибольших длин серий изучено в [14, 15], а также в [16, глава 4]. В [17] получены асимптотические формулы для вероятностей цепочек в последовательности независимых случайных величин и проведено обобщение этого результата для марковских последовательностей. В [18, 19] доказаны предельные теоремы Пуассона для числа повторений и кратных повторений (соответственно) цепочек в эргодической марковской цепи с оценками скорости сходимости. В [20] доказана многомерная центральная предельная теорема для чисел серий в марковской случайной последовательности с двумя состояниями.

В последнее время особый интерес представляет анализ свойств скрытых цепей Маркова в связи с задачами распознавания образов, машинного обучения и анализа текста [21, 22]. Основные свойства скрытых марковских моделей подробно описаны в [23, главы 1–3]. Для таких моделей также представляет интерес задача о точном и асимптотическом поведении чисел серий и цепочек как в наблюдаемой последовательности, так и образованной состояниями цепи Маркова. В [24] приведён способ вычисления вероятностей появлений цепочек специального вида в последовательности, образованной состояниями скрытой марковской цепи, по известному участку наблю-

даемой последовательности. Однако вычисления по приведённым в [24] алгоритмам являются весьма трудоёмкими для больших длин последовательностей.

В настоящей работе рассматривается один частный случай скрытой марковской цепи, а именно полиномиальная последовательность, управляемая цепью Маркова. Такая последовательность может трактоваться как последовательность, полученная укрупнением состояний марковской цепи, и как скрытая марковская цепь. В [25] изучено предельное распределение для числа пар совпавших знаков в такой последовательности, когда её длина стремится к бесконечности, а наибольшая вероятность появления любого знака стремится к нулю (согласованным образом). В частности, доказаны пуассоновская и нормальная предельные теоремы (с оценками скорости сходимости) для чисел пар совпадений знаков и ряда связанных с ними случайных величин.

Данная работа посвящена изучению асимптотических свойств совместных распределений чисел серий разных знаков и различных длин в полиномиальной случайной последовательности, управляемой цепью Маркова с конечным числом состояний. Получены предельные теоремы пуассоновского и нормального типа для чисел серий длины не меньше заданной.

1. Оценки расстояния по вариации и предельные теоремы

Пусть $\mathbf{Z} = (Z_0, Z_1, Z_2, \dots, Z_T, \dots)$ — стационарная цепь Маркова с множеством состояний $E_M = \{1, \dots, M\}$ и стационарными вероятностями $\tilde{\pi}_k = \pi_k(n) = \mathsf{P}\{Z_n = k\}$. Обозначим

$$\pi_{kl}^{(m)} = \mathsf{P}\{Z_{t+m} = l | Z_t = k\},\ k, l \in E_M,$$

вероятность перехода из состояния k в состояние l за m шагов, $m \in \mathbb{N}$. Для простоты обозначений будем писать π_{kl} вместо $\pi_{kl}^{(1)}$. Известно [26, часть 2, § 3], что существуют константы $C, \alpha > 0$, при которых

$$\max_{l \in E_M} |\pi_{lk}^{(n)} - \tilde{\pi}_k| \leqslant C\tilde{\pi}_k e^{-\alpha n}. \tag{1}$$

Пусть на множестве $A_N = \{1, \dots, N\}$ заданы M вероятностных распределений $(p_a^{(j)}, a \in A_N), j = 1, \dots, M$. Рассмотрим последовательность случайных величин X_0, X_1, …, X_T, …, принимающих значения из множества A_N с вероятностями

$$\mathsf{P}\{X_j = k\} = p_k^{(Z_j)},\quad k \in A_N,\ j = 0, 1, 2, \dots$$

Пусть $s \geqslant 1$. Обозначим $\nu_t^a = I\{X_{t-1} \neq a, X_t = \dots = X_{t+s-1} = a\}$ индикатор случайного события, состоящего в том, что в момент t началась серия из знаков a длины не меньше s (для краткости будем писать a-серия). Тогда сумма случайных индикаторов

$$\varsigma_s^a = \sum_{t=1}^{T} \nu_t^a \tag{2}$$

равна числу a-серий длины не меньше s с началом в последовательности $X_1, \dots, X_T$.

Математическое ожидание случайной величины ς_s^a обозначим $\lambda_s^a = \mathbf{E}\varsigma_s^a$. Оно определяется формулой $\lambda_s^a = \mathbf{E}\lambda_s^a(\mathbf{Z})$, где

$$\lambda_s^a(\mathbf{Z}) = \mathbf{E}(\varsigma_s^a | \mathbf{Z}) = \sum_{t=1}^{T} (1 - p_a^{(Z_{t-1})}) \prod_{j=t}^{t+s-1} p_a^{(Z_j)}. \tag{3}$$

Пусть $p_* = \min\limits_{a \in A_N} \{p_{a*}\}$, $p^* = \max\limits_{a \in A_N} \{p_a^*\}$, $p_a^* = \max\limits_{j=1,\dots,M} \{p_a^{(j)}\}$, $p_{a*} = \min\limits_{j=1,\dots,M} \{p_a^{(j)}\}$.

Лемма 1. Пусть $T > s \geqslant 1$, $a \in A_N$ — фиксированный знак. Тогда

$$\mathbf{E}\varsigma_s^a = \lambda_s^a = T \sum_{k_0,\ldots,k_s \in E_M} (1 - p_a^{(k_0)})\tilde{\pi}_{k_0} \prod_{j=1}^{s} p_a^{(k_j)} \pi_{k_{j-1}k_j}; \tag{4}$$

$$\mathbf{D}\lambda_s^a(\mathbf{Z}) \leqslant \lambda_s^a \left((2s+1)(1-p_{a*})(p_a^*)^s + \frac{2C\lambda_s^a}{T(e^\alpha - 1)} \right). \tag{5}$$

Замечание 1. Формула, аналогичная (4), приведена в [27] для более общего случая, когда цепь $\mathbf{Z}$ не является стационарной.

Пусть $\mathcal{A} \subset A_N$ — фиксированное подмножество и заданы наборы чисел $(s_a, a \in \mathcal{A})$ и $\lambda_{\mathcal{A}} = (\lambda_{s_a}^a, a \in \mathcal{A})$. Обозначим вектор $\varsigma_{\mathcal{A}} = (\varsigma_{s_a}^a, a \in \mathcal{A})$. Через $\mathrm{Pois}(\lambda_{\mathcal{A}})$ будем обозначать распределение случайного вектора с независимыми компонентами, каждая из которых распределена по закону Пуассона с параметром $\lambda_{s_a}^a$. Будем также использовать обозначение $\mathcal{L}(X)$ для распределения случайной величины X.

Напомним, что расстояние по вариации между распределениями случайных величин η_1 и η_2, принимающих значения в счётном множестве $\mathcal{E}$, выражается формулой

$$\rho_{TV}\left(\mathcal{L}(\eta_1), \mathcal{L}(\eta_2)\right) = \frac{1}{2} \sum_{a \in \mathcal{E}} |\mathsf{P}\{\eta_1 = a\} - \mathsf{P}\{\eta_2 = a\}|. \tag{6}$$

Пусть $s^* = \max\limits_{a \in \mathcal{A}}\{s_a\}$, $s_* = \min\limits_{a \in \mathcal{A}}\{s_a\}$. Мощность множества $\mathcal{B}$ будем обозначать $|\mathcal{B}|$.

Теорема 1. Пусть $\mathcal{A} \subseteq A_N$, задан набор натуральных чисел $(s_a, a \in \mathcal{A})$, $T \geqslant \geqslant \dfrac{C\lambda_{s_a}^a}{(e^\alpha - 1)(p_a^*)^{s_a}}$. Тогда выполнена оценка

$$\rho_{TV}\left(\mathcal{L}(\varsigma_{\mathcal{A}}), \mathrm{Pois}(\lambda_{\mathcal{A}})\right) \leqslant \gamma\left(\gamma T (p^*)^{s_*} + 1\right), \tag{7}$$

где $\gamma^2 = |\mathcal{A}|^2(2s^* + 3)(p^*)^{s_*}$.

Замечание 2. Оценка расстояния по вариации, аналогичная оценке (7), для случая одноточечного множества $\mathcal{A}$ получена в [28]. Если $\mathcal{A}$ — одноточечное множество, то вектор $\varsigma_{\mathcal{A}}$ представляет собой одномерную случайную величину. В случае одномерного распределения оценка работы [28] точнее, так как при доказательстве использован одномерный вариант метода Чена — Стейна.

Следствие 1. Пусть $\mathcal{A} \subseteq A_N$ — фиксированное множество, $p^* \in (0,1)$, $T \to \infty$, а числа $s_a, a \in \mathcal{A}$, меняются так, что $s_a \to \infty$ и $T(p^*)^{s_*} \to c > 0$. Тогда при всех $a \in \mathcal{A}$ существуют

$$\lim \mathbf{E}\varsigma_{s_a}^a = \lim T \sum_{k_0,\ldots,k_s \in E_M} (1 - p_a^{(k_0)})\tilde{\pi}_{k_0} \prod_{j=1}^{s} p_a^{(k_j)} \pi_{k_{j-1}k_j} = \lambda_a \geqslant 0,$$

а компоненты случайного вектора $\varsigma_{\mathcal{A}} = (\varsigma_{s_a}^a, a \in \mathcal{A})$ асимптотически независимы и имеют в пределе распределения Пуассона с параметрами λ_a соответственно.

Следствие 2. Пусть $\mathcal{A} \subseteq A_N$ — фиксированное множество, $p^* \in (0,1)$, $T \to \infty$, а числа $s_a, a \in \mathcal{A}$, меняются так, что $T(p^*)^{s_*} \to \infty$ и $\gamma T(p^*)^{s_*} \to 0$. Тогда компоненты случайного вектора $\left(\dfrac{\varsigma_{s_a}^a - \lambda_{s_a}^a}{\sqrt{\mathbf{D}\varsigma_{s_a}^a}}, a \in \mathcal{A} \right)$ асимптотически независимы и распределены в пределе по стандартному нормальному закону.

Замечание 3. Следствия 1 и 2 очевидным образом получаются из оценки (7), поэтому подробно на их доказательстве мы не останавливаемся.

2. Вывод оценки расстояния по вариации

Доказательство теоремы 1 начнём с более простой постановки задачи. Пусть случайные величины $Y_1, \ldots, Y_T, \ldots$ независимы и принимают значения из множества $A_N = \{1, \ldots, N\}$, причём

$$\mathsf{P}\{Y_j = k\} = p_k^{(j)}, \quad k \in A_N, \ j \in \{1, \ldots, M\},$$

а наборы $\{p_k^{(j)}\}$ при каждом j удовлетворяют условию $\sum\limits_{k \in A_N} p_k^{(j)} = 1$. Классу таких последовательностей принадлежит последовательность $X_0, \ldots, X_T, \ldots$, описанная в начале работы, при фиксированной $\mathbf{Z}$.

Пусть $s \geqslant 1$, $\tilde{\nu}_t^a = I\{Y_{t-1} \neq a, Y_t = \ldots = Y_{t+s-1} = a\}$, $\tilde{\zeta}_s^a = \sum\limits_{t=1}^{T} \tilde{\nu}_t^a$. Случайные величины $\tilde{\nu}_t^a$ и $\tilde{\zeta}_s^a$ аналогичны введённым в начале работы ν_t^a и ζ_s^a (см. (2)). Математическое ожидание случайной величины $\tilde{\zeta}_s^a$ определяется формулой

$$\mathbf{E}\tilde{\zeta}_s^a = \sum_{t=1}^{T} \mathbf{E}\tilde{\nu}_t^a = \sum_{t=1}^{T} (1 - p_a^{(t-1)}) \prod_{j=t}^{t+s-1} p_a^{(j)}. \tag{8}$$

Обозначим вектор $\tilde{\zeta}_{\mathcal{A}} = \left(\tilde{\zeta}_{s_a}^a, a \in \mathcal{A}\right)$. Сопровождающим пуассоновским распределением для вектора $\tilde{\zeta}_{\mathcal{A}}$ будет распределение вектора с независимыми компонентами, каждая из которых распределена по закону Пуассона с параметром $\mathbf{E}\tilde{\zeta}_{s_a}^a$, $a \in \mathcal{A}$, определяемым формулой (8). Для этого распределения будем использовать обозначение $\mathrm{Pois}(\mathbf{E}\tilde{\zeta}_{\mathcal{A}}) = \left(\mathrm{Pois}(\mathbf{E}\tilde{\zeta}_{s_a}^a), a \in \mathcal{A}\right)$.

Лемма 2. Пусть $\mathcal{A} \subseteq A_N$, задан набор натуральных чисел $s_a, a \in \mathcal{A}$, $T \geqslant 1$. Тогда выполнена оценка

$$\rho_{TV}\left(\mathcal{L}(\tilde{\zeta}_{\mathcal{A}}), \mathrm{Pois}(\mathbf{E}\tilde{\zeta}_{\mathcal{A}})\right) \leqslant (2s^* + 3)|\mathcal{A}|(p^*)^{s_*}\Lambda_{\mathcal{A}}, \tag{9}$$

где $\Lambda_{\mathcal{A}} = \sum\limits_{a \in \mathcal{A}} \mathbf{E}\tilde{\zeta}_{s_a}^a$.

Замечание 4. Если цепь Маркова $\mathbf{Z}$ представляет собой последовательность независимых одинаково распределённых случайных величин, то $X_0, X_1, \ldots, X_T, \ldots$ является стационарной последовательностью с полиномиальным распределением. В этом случае оценка (9) может быть получена из теоремы 2 работы [2].

Доказательство леммы 2 приведено далее в п. 3. Вернёмся к задаче об управляемой последовательности. Запишем неравенство треугольника

$$\rho_{TV}\left(\mathcal{L}(\zeta_{\mathcal{A}}), \mathrm{Pois}(\lambda_{\mathcal{A}})\right) \leqslant \rho_{TV}\left(\mathcal{L}(\zeta_{\mathcal{A}}), \mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z}))\right) + \rho_{TV}\left(\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z})), \mathrm{Pois}(\lambda_{\mathcal{A}})\right). \tag{10}$$

Здесь обозначение $\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z}))$ используется для распределения вектора, компоненты которого имеют смешанные пуассоновские распределения с параметрами $\lambda_{s_a}^a(\mathbf{Z})$. Напомним, что случайная величина X имеет смешанное распределение Пуассона с дискретным случайным параметром Λ, принимающим значения в множестве $\mathcal{E}$, если

$$\mathsf{P}\{X = n\} = \sum_{\lambda \in \mathcal{E}} \frac{\lambda^n}{n!} e^{-\lambda} \mathsf{P}\{\Lambda = \lambda\}.$$

Для оценки первого слагаемого в (10) воспользуемся леммой 2, согласно которой для каждой траектории $\mathbf{z}$ цепи Маркова $\mathbf{Z}$

$$\rho_{TV}\left(\mathcal{L}(\zeta_{\mathcal{A}}), \mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{z}))\right) \leqslant (2s^* + 3)|\mathcal{A}|(p^*)^{s_*}\Lambda_{\mathcal{A}}(\mathbf{z}); \tag{11}$$

$$\Lambda_{\mathcal{A}}(\mathbf{z})=\sum_{a\in\mathcal{A}}\lambda^a_{s_a}(\mathbf{z})=\sum_{a\in\mathcal{A}}\sum_{t=1}^{T}(1-p_a^{(z_{t-1})})\prod_{j=t}^{t+s_a-1}p_a^{(z_j)}\leqslant T|\mathcal{A}|(1-p_*)(p^*)^{s_*}\leqslant T|\mathcal{A}|(p^*)^{s_*}, \quad (12)$$

а также следующим утверждением.

Лемма 3. Пусть случайная величина Π имеет смешанное дискретное распределение со случайным параметром Θ. Обозначим β не зависящую от Π и Θ случайную величину. Тогда

$$\rho_{TV}\left(\mathcal{L}(\Pi),\mathcal{L}(\alpha)\right)\leqslant\max_{\theta}\rho_{TV}(\mathcal{L}(\Pi|\theta),\mathcal{L}(\beta)), \quad (13)$$

где максимум берется по всем возможным значениям Θ, а через $\mathcal{L}(\Pi|\theta)$ обозначено распределение случайной величины Π при фиксированном значении параметра $\Theta=\theta$.

Из (11)–(13) следует, что

$$\rho_{TV}\left(\mathcal{L}(\varsigma_{\mathcal{A}}),\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z}))\right)\leqslant 2T(s^*+1)|\mathcal{A}|^2(p^*)^{2s_*}=\gamma^2T(p^*)^{s_*}. \quad (14)$$

Для оценки второго слагаемого в правой части (10) понадобится ещё одно утверждение.

Лемма 4. В условиях теоремы 1

$$\rho_{TV}\left(\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z})),\mathrm{Pois}(\lambda_{\mathcal{A}})\right)\leqslant\gamma. \quad (15)$$

Подставляя (14) и (15) в (10), получаем

$$\rho_{TV}\left(\mathcal{L}(\varsigma_{\mathcal{A}}),\mathrm{Pois}(\lambda_{\mathcal{A}})\right)\leqslant\gamma^2T(p^*)^{s_*}+\gamma=\gamma\left(\gamma T(p^*)^{s_*}+1\right).$$

Теорема 1 доказана. ■

3. Доказательства лемм

Доказательство леммы 1

Формула (4) в нашем случае очевидна (см. замечание к лемме 1), поэтому сразу перейдём к формуле (5).

Вычислим сначала второй момент случайной величины $\lambda^a_s(\mathbf{Z})$. Согласно (3),

$$\begin{aligned}\mathbf{E}(\lambda^a_s(\mathbf{Z}))^2&=\mathbf{E}\left(\sum_{t=1}^{T}(1-p_a^{(Z_{t-1})})\prod_{j=t}^{t+s-1}p_a^{(Z_j)}\right)^2=\\&=\mathbf{E}\sum_{t=1}^{T}\sum_{t'=1}^{T}\left(1-p_a^{(Z_{t-1})}\right)\prod_{j=t}^{t+s-1}p_a^{(Z_j)}\left(1-p_a^{(Z_{t'-1})}\right)\prod_{j'=t'}^{t'+s-1}p_a^{(Z_{j'})}=\\&=\mathbf{E}\sum_{t=1}^{T}\left(1-p_a^{(Z_{t-1})}\right)^2\prod_{j=t}^{t+s-1}\left(p_a^{(Z_j)}\right)^2+\\&+\mathbf{E}\sum_{t=1}^{T}\sum_{\substack{t'=1,\dots,T;\\ t'\neq t}}\left(1-p_a^{(Z_{t-1})}\right)\prod_{j=t}^{t+s-1}p_a^{(Z_j)}\left(1-p_a^{(Z_{t'-1})}\right)\prod_{j'=t'}^{t'+s-1}p_a^{(Z_{j'})}.\end{aligned} \quad (16)$$

Оценим отдельно слагаемые в правой части (16). Для первого слагаемого имеем

$$\begin{aligned}\mathbf{E}\sum_{t=1}^{T}\left(1-p_a^{(Z_{t-1})}\right)^2\prod_{j=t}^{t+s-1}\left(p_a^{(Z_j)}\right)^2&\leqslant(1-p_{a*})(p_a^*)^s\mathbf{E}\sum_{t=1}^{T}\left(1-p_a^{(Z_{t-1})}\right)\prod_{j=t}^{t+s-1}p_a^{(Z_j)}\leqslant\\&\leqslant(1-p_{a*})(p_a^*)^s\lambda^a_s.\end{aligned} \quad (17)$$

Перейдём к оцениванию второго слагаемого. Пусть $t' \in \{t+1, \dots, t+s\}$. Тогда аналогично предыдущему случаю

$$\begin{aligned} &\mathbf{E}\left(1-p_a^{(Z_{t-1})}\right)\prod_{j=t}^{t+s-1} p_a^{(Z_j)}\left(1-p_a^{(Z_{t'-1})}\right)\prod_{j'=t'}^{t'+s-1} p_a^{(Z_{j'})} \leqslant \\ &\leqslant \mathbf{E}\left(1-p_a^{(Z_{t-1})}\right)\prod_{j=t}^{t+s-1} p_a^{(Z_j)}(1-p_{a*})(p_a^*)^s. \end{aligned} \tag{18}$$

Пусть $t' \geqslant t+s+1$. Тогда

$$\begin{aligned} &\mathbf{E}(1-p_a^{(Z_{t-1})})\prod_{j=t}^{t+s-1} p_a^{(Z_j)}(1-p_a^{(Z_{t'-1})})\prod_{j'=t'}^{t'+s-1} p_a^{(Z_{j'})} = \\ &= \sum_{k_0,\dots,k_s\in E_M}\sum_{l_0,\dots,l_s\in E_M} \mathsf{P}\{Z_{t-1}=k_0,\dots,Z_{t+s-1}=k_s, Z_{t'-1}=l_0\dots,Z_{t'+s-1}=l_s\}\times \\ &\times(1-p_a^{(k_0)})\prod_{j=1}^{s} p_a^{(k_j)}(1-p_a^{(l_0)})\prod_{j'=1}^{s} p_a^{(l_{j'})} = \\ &= \sum_{k_0,\dots,k_s\in E_M}\tilde\pi_{k_0}(1-p_a^{(k_0)})\prod_{j=1}^{s} p_a^{(k_j)}\pi_{k_{j-1}k_j}\sum_{l_0,\dots,l_s\in E_M}(1-p_a^{(l_0)})\pi_{k_s l_0}^{(t'-t-s)}\prod_{j'=1}^{s} p_a^{(l_{j'})}\pi_{l_{j'-1}l_{j'}} \leqslant \\ &\leqslant \sum_{k_0,\dots,k_s\in E_M}\tilde\pi_{k_0}(1-p_a^{(k_0)})\prod_{j=1}^{s} p_a^{(k_j)}\pi_{k_{j-1}k_j}\sum_{l_0,\dots,l_s\in E_M}(1-p_a^{(l_0)})\tilde\pi_{l_0}(1+Ce^{-\alpha(t'-t-s)})\prod_{j'=1}^{s} p_a^{(l_{j'})}\pi_{l_{j'-1}l_{j'}} \end{aligned}$$

(в последнем неравенстве мы воспользовались оценкой (1)). Значит,

$$\begin{aligned} &\mathbf{E}\sum_{t=1}^{T}\sum_{\substack{t'=1,\dots,T;\\ |t'-t|>s}}(1-p_a^{(Z_{t-1})})\prod_{j=t}^{t+s-1} p_a^{(Z_j)}(1-p_a^{(Z_{t'-1})})\prod_{j'=t'}^{t'+s-1} p_a^{(Z_{j'})} \leqslant \\ &\leqslant \sum_{t=1}^{T}\sum_{\substack{t'=1,\dots,T;\\ |t'-t|>s}}\sum_{k_0,\dots,k_s\in E_M}\tilde\pi_{k_0}(1-p_a^{(k_0)})\prod_{j=1}^{s} p_a^{(k_j)}\pi_{k_{j-1}k_j}\times \\ &\times\sum_{l_0,\dots,l_s\in E_M}(1-p_a^{(l_0)})\tilde\pi_{l_0}(1+Ce^{-\alpha|t'-t-s|})\prod_{j'=1}^{s} p_a^{(l_{j'})}\pi_{l_{j'-1}l_{j'}}. \end{aligned} \tag{19}$$

Так как $\sum\limits_{\substack{t'=1,\dots,T;\\ |t'-t|>s}} e^{-\alpha|t'-t-s|} = \dfrac{1-e^{\alpha(s+t-T)}}{e^\alpha-1}+\dfrac{1-e^{\alpha(1-(s+t))}}{e^\alpha-1} \leqslant \dfrac{2}{e^\alpha-1}$, то правая часть (19) оценивается сверху выражением

$$\begin{aligned} &\sum_{t=1}^{T}\sum_{k_0,\dots,k_s\in E_M}\tilde\pi_{k_0}(1-p_a^{(k_0)})\prod_{j=1}^{s} p_a^{(k_j)}\pi_{k_{j-1}k_j}\times \\ &\times\sum_{l_0,\dots,l_s\in E_M}(1-p_a^{(l_0)})\tilde\pi_{l_0}\left(T+\frac{2C}{e^\alpha-1}\right)\prod_{j'=1}^{s} p_a^{(l_{j'})}\pi_{l_{j'-1}l_{j'}} = \\ &= T\sum_{k_0,\dots,k_s\in E_M}\tilde\pi_{k_0}(1-p_a^{(k_0)})\prod_{j=1}^{s} p_a^{(k_j)}\pi_{k_{j-1}k_j}\times \\ &\times T\sum_{l_0,\dots,l_s\in E_M}(1-p_a^{(l_0)})\tilde\pi_{l_0}\left(1+\frac{2C}{T(e^\alpha-1)}\right)\prod_{j'=1}^{s} p_a^{(l_{j'})}\pi_{l_{j'-1}l_{j'}} = \\ &= (\lambda_s^a)^2\left(1+\frac{2}{C}T(e^\alpha-1)\right). \end{aligned} \tag{20}$$

Подставив оценки (17)–(20) в (16), получим

$$\mathbf{E}(\lambda_s^a(\mathbf{Z}))^2 \leqslant \lambda_s^a\left((2s+1)(1-p_{a*})(p_a^*)^s + \lambda_s^a\left(1+\frac{2C}{T(e^\alpha-1)}\right)\right),$$
$$\mathbf{D}\lambda_s^a(\mathbf{Z}) \leqslant \lambda_s^a\left((2s+1)(1-p_{a*})(p_a^*)^s + \frac{2C\lambda_s^a}{T(e^\alpha-1)}\right).$$

Лемма 1 доказана. ■

Доказательство леммы 2

Для каждой пары индексов (a,t), $t \in \{1,\dots,T\}$, $a \in A_N$, определим множество $O(a,t)$ равенством

$$O(a,t) = \{(b,t') : b \in \mathcal{A}, t' \in \{1,\dots,T\} : \max\{1, t-s_b\} \leqslant t' \leqslant \min\{T, t+s_a\}\}.$$

Тогда случайный индикатор $\tilde\nu_t^a$ и набор случайных индикаторов $\left(\tilde\nu_{t'}^b\right)$, $(b,t') \notin O(a,t)$, независимы. Согласно [29, глава 10, с. 210, теорема 10.A], расстояние по вариации между распределением случайного вектора $\tilde\zeta_{\mathcal{A}} = \left(\tilde\zeta_{s_a}^a, a \in \mathcal{A}\right)$ и сопровождающим пуассоновским распределением $\mathrm{Pois}(\mathbf{E}\tilde\zeta_{\mathcal{A}}) = \left(\mathrm{Pois}(\mathbf{E}\tilde\zeta_{s_a}^a), a \in \mathcal{A}\right)$ оценивается как

$$\rho_{TV}(\mathcal{L}(\tilde\zeta_{\mathcal{A}}), \mathrm{Pois}(\mathbf{E}\tilde\zeta_{\mathcal{A}})) \leqslant S_1 + S_2, \tag{21}$$

$$S_1 = \sum_{a\in\mathcal{A}}\sum_{t=1}^{T}\sum_{(b,t')\in O(a,t)} \mathbf{E}\tilde\nu_t^a\mathbf{E}\tilde\nu_{t'}^b, \quad S_2 = \sum_{a\in\mathcal{A}}\sum_{t=1}^{T}\sum_{(b,t')\in O(a,t)\setminus\{(a,t\}\}} \mathbf{E}\tilde\nu_t^a\tilde\nu_{t'}^b.$$

Начнём с оценивания первой суммы S_1. Так как

$$\mathbf{E}\tilde\nu_{t'}^b = (1-p_b^{(t'-1)})\prod_{j=t'}^{t'+s_b-1} p_b^{(j)} \leqslant (1-p_*)\prod_{j=t'}^{t'+s_b-1} p^* \leqslant (1-p_*)(p^*)^{s_*}, \tag{22}$$

то

$$\begin{aligned} S_1 &= \sum_{a\in\mathcal{A}}\sum_{1\leqslant t\leqslant T}\mathbf{E}\tilde\nu_t^a\sum_{(b,t')\in O(a,t)}\mathbf{E}\tilde\nu_{t'}^b \leqslant (2s^*+1)|\mathcal{A}|(1-p_*)(p^*)^{s_*}\sum_{a\in\mathcal{A}}\sum_{1\leqslant t\leqslant T}\mathbf{E}\tilde\nu_t^a = \\ &= (2s^*+1)|\mathcal{A}|(1-p_*)(p^*)^{s_*}\sum_{a\in\mathcal{A}}\mathbf{E}\tilde\zeta_{s_a}^a = (2s^*+1)|\mathcal{A}|(1-p_*)(p^*)^{s_*}\Lambda_{\mathcal{A}} \end{aligned} \tag{23}$$

(в силу формулы (8)).

Теперь перейдём к оцениванию S_2. Перепишем формулу для S_2 в виде

$$S_2 = \sum_{a,b\in\mathcal{A}} S_2(a,b), \quad S_2(a,b) = \sum_{t=1}^{T}\sum_{(b,t')\in O(a,t)\setminus\{(a,t\}\}}\mathbf{E}\tilde\nu_t^a\tilde\nu_{t'}^b.$$

Заметим, что события, соответствующие случайным индикаторам $\tilde\nu_t^a$ и $\tilde\nu_{t'}^a$, $(a,t') \in$ $\in O(a,t)\setminus\{(a,t)\}$, несовместны. Действительно, при $t' > t$ события $\{Y_{t-1} \neq a, Y_t = \dots = Y_{t+s_a-1} = a\}$ и $\{Y_{t'-1} \neq a, Y_{t'} = \dots = Y_{t'+s_a-1} = a\}$ (отвечающие случайным индикаторам $\tilde\nu_t^a$ и $\tilde\nu_{t'}^a$ соответственно) зависят от одного и того же случайного знака $Y_{t'-1}$, который в первом событии должен принимать значение a, а во втором — любое значение, отличное от a. Аналогично при $t' < t$. Поэтому $S_2(a,a) = 0$.

Пусть теперь $a \neq b$. События, отвечающие $\tilde\nu_t^a$ и $\tilde\nu_{t'}^b$, $(b,t') \in O(a,t)\setminus\{(a,t)\}$, совместны, если $t' = t - s_b$ или $t' = t + s_a$ (аналогично оценке для $S_2(a,a)$), и

$$\mathbf{E}\tilde\nu_t^a\tilde\nu_{t-s_b}^b = \mathbf{E}\tilde\nu_t^a\tilde\nu_{t+s_a}^b = \frac{1}{\mathsf{P}\{Y_{t-1}\neq a\}}\mathbf{E}\tilde\nu_t^a\mathbf{E}\tilde\nu_{t-s_b}^b \leqslant (1-p_*)^{-1}\mathbf{E}\tilde\nu_t^a\mathbf{E}\tilde\nu_{t-s_b}^b.$$

Поэтому из (22) имеем

$$S_2(a,b) = 2\sum_{t=1}^{T}(1-p_*)^{-1}\mathbf{E}\tilde{\nu}_t^a\mathbf{E}\tilde{\nu}_{t-s_b}^b \leqslant 2(p^*)^{s_*}\sum_{1\leqslant t\leqslant T}\mathbf{E}\tilde{\nu}_t^a = 2(p^*)^{s_*}\mathbf{E}\tilde{\varsigma}_{s_a}^a.$$

Таким образом,

$$S_2 = \sum_{a,b\in\mathcal{A}} S_2(a,b) \leqslant 2(p^*)^{s_*}\sum_{a,b\in\mathcal{A}}\mathbf{E}\tilde{\varsigma}_{s_a}^a \leqslant 2|\mathcal{A}|(p^*)^{s_*}\sum_{a\in\mathcal{A}}\mathbf{E}\tilde{\varsigma}_{s_a}^a = 2|\mathcal{A}|(p^*)^{s_*}\Lambda_{\mathcal{A}}. \tag{24}$$

Подставляя оценки (23) и (24) в (21), получаем

$$\rho_{TV}(\mathcal{L}(\tilde{\zeta}_{\mathcal{A}}), \mathrm{Pois}(\mathbf{E}\tilde{\zeta}_{\mathcal{A}})) \leqslant (2s^*+3)|\mathcal{A}|(p^*)^{s_*}\Lambda_{\mathcal{A}}.$$

Лемма 2 доказана. ■

Доказательство леммы 4

Нам понадобятся ещё одно вспомогательное утверждение и несколько дополнительных определений.

Пусть случайные величины Π_j, $j = 1,\dots,k$, $k \geqslant 1$, имеют смешанные дискретные распределения с параметрами $\Theta_j = \Theta_j(\xi)$. Будем писать $\mathcal{L}(\Pi_j|\theta_j)$ для обозначения распределения случайной величины Π_j при фиксированном значении параметра $\Theta_j = \theta_j$.

Пусть $\alpha_1,\dots,\alpha_r$ — дискретные случайные величины. Обозначим при каждом фиксированном значении $\Theta_j = \theta_j$

$$r(\mathcal{L}(\Pi_j), \mathcal{L}(\alpha_j)) = \rho_{TV}(\mathcal{L}(\Pi_j|\theta_j), \mathcal{L}(\alpha_j)).$$

Величина $r(\mathcal{L}(\Pi_j), \mathcal{L}(\alpha_j))$ равна расстоянию по вариации (см. (6)) между распределениями $\mathcal{L}(\Pi_j|\theta_j)$ и $\mathcal{L}(\alpha_j)$.

Лемма 5. Пусть ξ — дискретная случайная величина, случайные величины Π_j, $j = 1,\dots,k$, $k \geqslant 1$, имеют смешанные дискретные распределения с параметрами $\Theta_j = \Theta_j(\xi)$, причём при каждом фиксированном $\xi = x$ случайные величины Π_j, $j = 1,\dots,k$, независимы. Обозначим $\alpha_1,\dots,\alpha_k$ независимые и не зависящие от Π_j, $j = 1,\dots,k$, случайные величины. Тогда

$$\rho_{TV}\left(\mathcal{L}(\Pi_1,\dots,\Pi_k), \mathcal{L}(\alpha_1,\dots,\alpha_k)\right) \leqslant \sum_{j=1}^{k}\mathbf{E}r(\mathcal{L}(\Pi_j), \mathcal{L}(\alpha_j)). \tag{25}$$

Из (25), подставляя вместо $(\Pi_1,\dots,\Pi_k)$ вектор со смешанным пуассоновским распределением $\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z}))$, а вместо $\alpha_1,\dots,\alpha_k$ — вектор с распределением $\mathrm{Pois}(\lambda_{\mathcal{A}})$, получаем

$$\rho_{TV}\left(\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z})), \mathrm{Pois}(\lambda_{\mathcal{A}})\right) \leqslant \sum_{a\in\mathcal{A}}\mathbf{E}r\left(\mathrm{Pois}(\lambda_{s_a}^a(\mathbf{Z})), \mathrm{Pois}(\lambda_{s_a}^a)\right). \tag{26}$$

Воспользуемся теоремой 1.C из [29, глава 1, с. 12], которая в нашем случае формулируется следующим образом:

$$\rho_{TV}\left(\mathrm{Pois}(\lambda), \mathrm{Pois}(\mu)\right) \leqslant \min\left\{1, \frac{1}{\sqrt{\lambda}}, \frac{1}{\sqrt{\mu}}\right\}|\lambda-\mu|.$$

Тогда

$$\begin{gathered}\mathbf{E}r\left(\mathrm{Pois}(\lambda^a_{s_a}(\mathbf{Z})),\mathrm{Pois}(\lambda^a_{s_a})\right)\leqslant\min\left\{1,\frac{1}{\sqrt{\lambda^a_{s_a}}}\right\}\mathbf{E}|\lambda^a_{s_a}(\mathbf{Z})-\lambda^a_{s_a}|\leqslant\\ \leqslant\min\left\{1,\frac{1}{\sqrt{\lambda^a_{s_a}}}\right\}\sqrt{\mathbf{D}\lambda^a_{s_a}(\mathbf{Z})}.\end{gathered}\tag{27}$$

Из (5) и (27) следует, что

$$\mathbf{E}r\left(\mathrm{Pois}(\lambda^a_{s_a}(\mathbf{Z})),\mathrm{Pois}(\lambda^a_{s_a})\right)\leqslant\sqrt{(2s_a+1)(1-p_{a*})(p^*_a)^s+\frac{2C\lambda^a_{s_a}}{T(e^\alpha-1)}}.$$

Значит, из последней оценки и формулы (26) получаем, что при T, удовлетворяющих неравенству $\dfrac{C\lambda^a_{s_a}}{T(e^\alpha-1)}\leqslant(p^*_a)^{s_a}$, имеет место оценка

$$\rho_{TV}\left(\mathrm{Pois}(\lambda_{\mathcal{A}}(\mathbf{Z})),\mathrm{Pois}(\lambda_{\mathcal{A}})\right)\leqslant\sum_{a\in\mathcal{A}}\sqrt{(2s_a+3)(p^*_a)^{s_a}}\leqslant|\mathcal{A}|\sqrt{(2s^*+3)}(p^*)^{s_*/2}=\gamma.$$

Лемма 4 доказана. ■

Доказательство леммы 5

Последовательно применяя неравенство треугольника, получаем следующую цепочку неравенств:

$$\begin{gathered}\rho_{TV}\left(\mathcal{L}(\Pi_1,\dots,\Pi_k),\mathcal{L}(\alpha_1,\dots,\alpha_k)\right)\leqslant\rho_{TV}\left(\mathcal{L}(\Pi_1,\Pi_2,\dots,\Pi_k),\mathcal{L}(\alpha_1,\Pi_2,\dots,\Pi_k)\right)+\\ +\rho_{TV}\left(\mathcal{L}(\alpha_1,\Pi_2,\dots,\Pi_k),\mathcal{L}(\alpha_1,\dots,\alpha_k)\right)\leqslant\rho_{TV}\left(\mathcal{L}(\Pi_1,\Pi_2,\dots,\Pi_k),\mathcal{L}(\alpha_1,\Pi_2,\dots,\Pi_k)\right)+\\ +\rho_{TV}\left(\mathcal{L}(\alpha_1,\Pi_2,\Pi_3,\dots,\Pi_k),\mathcal{L}(\alpha_1,\alpha_2,\Pi_3,\dots,\Pi_k)\right)+\\ +\rho_{TV}\left(\mathcal{L}(\alpha_1,\alpha_2,\Pi_3,\dots,\Pi_k),\mathcal{L}(\alpha_1,\alpha_2,\dots,\alpha_k)\right)\leqslant\dots\leqslant\\ \leqslant\sum_{j=1}^{k}\rho_{TV}\left(\mathcal{L}(\alpha_1,\dots,\alpha_{j-1},\Pi_j,\dots,\Pi_k),\mathcal{L}(\alpha_1,\dots,\alpha_{j-1},\alpha_j,\Pi_{j+1},\dots,\Pi_k)\right)\end{gathered}\tag{28}$$

(в формуле (28) считаем, что слагаемое при $j=1$ в правой части неравенства равно $\rho_{TV}\left(\mathcal{L}(\Pi_1,\Pi_2,\dots,\Pi_k),\mathcal{L}(\alpha_1,\Pi_2,\dots,\Pi_k)\right)$).

Рассмотрим отдельно первое слагаемое в (28). Согласно определению расстояния по вариации (см. (6)), имеем

$$\begin{gathered}\rho_{TV}\left(\mathcal{L}(\Pi_1,\Pi_2,\dots,\Pi_k),\mathcal{L}(\alpha_1,\Pi_2,\dots,\Pi_k)\right)=\\ =\frac{1}{2}\sum_{l_1,\dots l_k=0}^{\infty}|\mathsf{P}\{\Pi_1=l_1,\Pi_2=l_2,\dots,\Pi_k=l_k\}-\mathsf{P}\{\alpha_1=l_1,\Pi_2=l_2,\dots,\Pi_k=l_k\}|.\end{gathered}$$

Так как параметры смешанных распределений случайных величин $\Pi_1,\Pi_2,\dots,\Pi_k$ зависят от дискретной случайной величины ξ, для вычисления оценки расстояния по вариации воспользуемся формулой полной вероятности:

$$\begin{gathered}\rho_{TV}\left(\mathcal{L}(\Pi_1,\Pi_2,\dots,\Pi_k),\mathcal{L}(\alpha_1,\Pi_2,\dots,\Pi_k)\right)=\frac{1}{2}\sum_{l_1,\dots l_k=0}^{\infty}\sum_x\mathsf{P}\{\xi=x\}\times\\ \times|\mathsf{P}\{\Pi_1=l_1,\Pi_2=l_2,\dots,\Pi_k=l_k|\xi=x\}-\mathsf{P}\{\alpha_1=l_1,\Pi_2=l_2,\dots,\Pi_k=l_k|\xi=x\}|=\\ =\frac{1}{2}\sum_{l_1,\dots l_k=0}^{\infty}\sum_x|\mathsf{P}\{\xi=x\}\left(\mathsf{P}\{\Pi_1=l_1,\Pi_2=l_2,\dots,\Pi_k=l_k|\xi=x\}-\right.\\ \left.-\mathsf{P}\{\alpha_1=l_1,\Pi_2=l_2,\dots,\Pi_k=l_k|\xi=x\}\right)|.\end{gathered}$$

Здесь и далее суммирование по индексу x означает, что суммирование ведётся по всем возможным значениям случайной величины ξ.

Так как при фиксированном значении $\xi = x$ случайные величины $\Pi_1, \Pi_2, \ldots, \Pi_k$ независимы, продолжая цепочку равенств, получаем

$$\begin{aligned}
&\rho_{TV}\left(\mathcal{L}(\Pi_1, \Pi_2, \ldots, \Pi_k), \mathcal{L}(\alpha_1, \Pi_2, \ldots, \Pi_k)\right) = \\
&= \frac{1}{2} \sum_{l_1,\ldots l_k=0}^{\infty} |\sum_x \mathsf{P}\{\xi = x\} \left(\mathsf{P}\{\Pi_1 = l_1 | \xi = x\} \mathsf{P}\{\Pi_2 = l_2, \ldots, \Pi_k = l_k | \xi = x\} - \right. \\
&\qquad \left. - \mathsf{P}\{\alpha_1 = l_1\} \mathsf{P}\{\Pi_2 = l_2, \ldots, \Pi_k = l_k | \xi = x\}\right)| = \\
&= \frac{1}{2} \sum_{l_1,\ldots l_k=0}^{\infty} |\sum_x \mathsf{P}\{\xi = x\} \mathsf{P}\{\Pi_2 = l_2, \ldots, \Pi_k = l_k | \xi = x\} \times \\
&\qquad \times \left(\mathsf{P}\{\Pi_1 = l_1 | \xi = x\} - \mathsf{P}\{\alpha_1 = l_1\}\right) | = \\
&= \frac{1}{2} \sum_{l_1,\ldots l_k=0}^{\infty} |\sum_x \mathsf{P}\{\xi = x\} \mathsf{P}\{\Pi_2 = l_2, \ldots, \Pi_k = l_k | \xi = x\} \times \\
&\qquad \times \left(\mathsf{P}\{\Pi_1 = l_1 | \xi = x\} - \mathsf{P}\{\alpha_1 = l_1\}\right)| \leqslant \\
&\leqslant \frac{1}{2} \sum_{l_1,\ldots l_k=0}^{\infty} \sum_x \mathsf{P}\{\xi = x\} \mathsf{P}\{\Pi_2 = l_2, \ldots, \Pi_k = l_k | \xi = x\} \times \\
&\qquad \times |\left(\mathsf{P}\{\Pi_1 = l_1 | \xi = x\} - \mathsf{P}\{\alpha_1 = l_1\}\right)| \leqslant \\
&\leqslant \frac{1}{2} \sum_{l_1=0}^{\infty} \sum_x \mathsf{P}\{\xi = x\} \, |\left(\mathsf{P}\{\Pi_1 = l_1 | \xi = x\} - \mathsf{P}\{\alpha_1 = l_1\}\right)| = \mathbf{E} r(\mathcal{L}(\Pi_1), \mathcal{L}(\alpha_1)).
\end{aligned} \tag{29}$$

Аналогично для слагаемых в (28) при $j \geqslant 2$:

$$\begin{aligned}
&\rho_{TV}\left(\mathcal{L}(\alpha_1, \ldots, \alpha_{j-1}, \Pi_j, \ldots, \Pi_k), \mathcal{L}(\alpha_1, \ldots, \alpha_{j-1}, \alpha_j, \Pi_{j+1}, \ldots, \Pi_k)\right) = \\
&= \frac{1}{2} \sum_{l_1,\ldots,l_k=0}^{\infty} |\mathsf{P}\{\alpha_1 = l_1, \ldots, \alpha_{j-1} = l_{j-1}, \Pi_j = l_j, \ldots, \Pi_k = l_k\} - \\
&\quad - \mathsf{P}\{\alpha_1 = l_1, \ldots, \alpha_{j-1} = l_{j-1}, \alpha_j = l_j, \Pi_{j+1} = l_{j+1}, \ldots, \Pi_k = l_k\}| = \\
&= \frac{1}{2} \sum_{l_1,\ldots,l_k=0}^{\infty} |\mathsf{P}\{\alpha_1 = l_1, \ldots, \alpha_{j-1} = l_{j-1}\} \mathsf{P}\{\Pi_j = l_j, \ldots, \Pi_k = l_k\} - \\
&\quad - \mathsf{P}\{\alpha_1 = l_1, \ldots, \alpha_{j-1} = l_{j-1}\} \mathsf{P}\{\alpha_j = l_j\} \mathsf{P}\{\Pi_{j+1} = l_{j+1}, \ldots, \Pi_k = l_k\}| \leqslant \\
&\leqslant \frac{1}{2} \sum_{l_1,\ldots,l_k=0}^{\infty} |\mathsf{P}\{\Pi_j = l_j, \ldots, \Pi_k = l_k\} - \mathsf{P}\{\alpha_j = l_j\} \mathsf{P}\{\Pi_{j+1} = l_{j+1}, \ldots, \Pi_k = l_k\}| \, .
\end{aligned}$$

Дальше достаточно применить доказанную ранее оценку (29):

$$\rho_{TV}\left(\mathcal{L}(\alpha_1, \ldots, \alpha_{j-1}, \Pi_j, \ldots, \Pi_k), \mathcal{L}(\alpha_1, \ldots, \alpha_{j-1}, \alpha_j, \Pi_{j+1}, \ldots, \Pi_k)\right) \leqslant \mathbf{E} r(\mathcal{L}(\Pi_j), \mathcal{L}(\alpha_j)).$$

Тогда из (28) и последнего неравенства получим

$$\rho_{TV}\left(\mathcal{L}(\Pi_1, \ldots, \Pi_k), \mathcal{L}(\alpha_1, \ldots, \alpha_k)\right) \leqslant \sum_{j=1}^{k} \mathbf{E} r(\mathcal{L}(\Pi_j), \mathcal{L}(\alpha_j)).$$

Лемма 5 доказана. ■

Доказательство леммы 3

Воспользуемся формулой (25) при $k = 1$:

$$\rho_{TV}\left(\mathcal{L}(\Pi_1), \mathcal{L}(\alpha_1)\right) \leqslant \mathbf{E} r(\mathcal{L}(\Pi_1), \mathcal{L}(\alpha_1)).$$

Так как $\mathbf{E} r(\mathcal{L}(\Pi_1), \mathcal{L}(\alpha_1)) \leqslant \max\limits_x r(\mathcal{L}(\Pi_1 | x_j), \mathcal{L}(\alpha_1))$, из двух последних формул получаем (13). ■

Заключение

Проведено исследование свойств распределения чисел серий в последовательности случайных величин с полиномиальными распределениями, управляемой стационарной цепью Маркова с конечным числом состояний. Использованы функциональный вариант метода Чена — Стейна, а также оценки для аппроксимации смешанного пуассоновского распределения простым распределением Пуассона. В работе получены:

1) оценка для расстояния по вариации между распределением случайного вектора из чисел серий заданных знаков и заданной длины в управляемой полиномиальной последовательности и сопровождающим многомерным распределением Пуассона;
2) выведены многомерные пуассоновская и нормальная предельные теоремы для указанного случайного вектора, когда длина T наблюдаемой последовательности стремится к бесконечности, а максимальная вероятность появления каждого знака — к нулю (согласованным образом с T).

Автор выражает глубокую признательность рецензенту за ценные замечания и рекомендации, а также В. Г. Михайлову за постановку задачи, советы и внимание к работе.

ЛИТЕРАТУРА

1. *Balakrishnan N. and Koutras M. V.* Runs and Scans with Applications. N. Y.: John Whiley & Sons Inc., 2002. 452 p.
2. *Михайлов В. Г.* Об асимптотических свойствах числа серий событий // Тр. по дискр. матем. 2006. Т. 9. С. 152–163.
3. *Aki S. and Hirano K.* Discrete distributions related to succession events in two-state Markov chain // Statistical Science and Data Analysis / eds. K. Matusita, M. L. Puri, T. Hayakawa. Zeist: VSP International Science Publishers, 1993. P. 467–474.
4. *Aki S. and Hirano K.* Sooner and later waiting time problems for runs in Markov dependent bivariate trials // Ann. Inst. Stat. Math. 1999. V. 51. P. 17–29.
5. *Han Q. and Aki S.* Formulae and recursions for the joint distributions of success runs of several lengths in a two-state Markov chain // Stat. Probab. Lett. 1998. V. 40. No. 3. P. 203–214.
6. *Савельев Л. Я., Балакин С. В.* Совместное распределение числа единиц и числа 1-серий в двоичных марковских последовательностях // Дискретная математика. 2004. Т. 16. № 3. С. 43–62.
7. *Савельев Л. Я., Балакин С. В.* Комбинаторное вычисление моментов характеристик серий в троичных марковских последовательностях // Дискретная математика. 2011. Т. 23. № 2. С. 76–92.
8. *Савельев Л. Я.* Распределения числа состояний в двоичных марковских стохастических моделях // Сиб. журн. вычисл. матем. 2015. Т. 18. № 2. С. 191–200.
9. *Савельев Л. Я., Балакин С. В.* Некоторые применения стохастической теории серий // Сиб. журн. индустр. матем. 2012. Т. 15. № 3. С. 111–123.
10. *Shinde R. L. and Kotwal K. S.* On the joint distribution of runs in the sequence of Markov-dependent multi-state trials // Stat. Probab. Lett. 2006. V. 76. No. 10. P. 1065–1074.
11. *Geske M. X., Godbole A. P., Schaffner A. A., et al.* Compound Poisson approximations for word patterns under Markovian hypotheses // J. Appl. Probab. 1995. V. 32. P. 877–892.
12. *Erhardsson T.* Compound Poisson approximation for Markov chains using Stein's method // Ann. Probab. 1999. V. 27. No. 1. P. 565–596.
13. *Chryssaphinou O. and Vaggelatou E.* Compound Poisson approximation for multiple runs in a Markov chain // Ann. Inst. Stat. Math. 2002. V. 54. No. 2. P. 411–424.

14. *Fu J. C., Wang L., and Lou W. Y. W.* On exact and large deviation approximation for the distribution of the longest run in a sequence of two-state Markov dependent trials // J. Appl. Probab. 2003. V. 40. No. 2. P. 346–360.
15. *Eryilmaz S.* Some results associated with the longest run statistic in a sequence of Markov dependent trials // Appl. Math. Comput. 2006. V. 175. No. 1. P. 119–130.
16. *Pinsky M. A. and Karlin S.* The long run behavior of Markov chains // An Introduction to Stochastic Modeling. Fourth Edition / eds. M. A. Pinsky, S. Karlin. Boston: Elsevier, 2011. P. 165–222.
17. *Fu J. C. and Johnson B. C.* Approximate probabilities for runs and patterns in i.i.d. and Markov-dependent multistate trials // Adv. Appl. Probab. Appl. Probab. Trust. 2009. V. 41. No. 1. P. 292–308.
18. *Михайлов В. Г., Шойтов А. М.* О длинных повторениях цепочек в цепи Маркова // Дискретная математика. 2014. Т. 26. № 3. С. 79–89.
19. *Михайлов В. Г.* Оценки точности пуассоновской аппроксимации для распределения числа серий повторений длинных цепочек в цепи Маркова // Дискретная математика. 2015. Т. 27. № 4. С. 67–78.
20. *Mytalas G. C. and Zazanis M. A.* Central Limit Theorem approximations for the number of runs in Markov-dependent binary sequences // J. Stat. Plan. Inference. 2013. V. 143. No. 2. P. 321–333.
21. *Mahmoudzadeh E., Montazeri M. A., Zekri M., and Sadri S.* Extended hidden Markov model for optimized segmentation of breast thermography images // Infrared Phys. Technol. 2015. V. 72. P. 19–28.
22. *Yang W., Tao J., and Ye Z.* Continuous sign language recognition using level building based on fast hidden Markov model // Pattern Recognit. Lett. 2016. V. 78. P. 28–35.
23. *Elliott R. J., Aggoun L., and Moore J. B.* Hidden Markov Models. N. Y.: Springer, 1995. V. 29. 382 p.
24. *Aston J. A. D. and Martin D. E. K.* Distributions associated with general runs and patterns in hidden Markov models // Ann. Appl. Stat. 2007. V. 1. No. 2. P. 585–611.
25. *Меженная Н. М.* О числе совпадений знаков в дискретной случайной последовательности, управляемой цепью Маркова // Сибирские электронные математические известия. 2016. Т. 13. С. 305–317.
26. *Розанов Ю. А.* Случайные процессы. Краткий курс. М.: Наука, 1979. 184 с.
27. *Mezhennaya N. M.* On the distribution of the number of runs in polynomial sequence controlled by Markov chain // OP&PM Surv. Appl. Ind. Math. 2016. V. 23. No. 2. P. 186–187.
28. *Меженная Н. М.* О предельном распределении числа серий в полиномиальной последовательности, управляемой цепью Маркова // Вестник УдГУ. 2016. Т. 26. № 3. С. 324–335.
29. *Barbour A. D., Holst L., and Janson S.* Poisson Approximation. Oxford: Oxford Univ. Press, 1992. 277 p.

REFERENCES

1. *Balakrishnan N. and Koutras M. V.* Runs and Scans with Applications. N. Y., John Whiley & Sons Inc., 2002, 452 p.
2. *Mikhaylov V. G.* Ob asimptoticheskikh svoystvakh chisla seriy sobytiy [On asymptotic properties of the number of runs of events]. Tr. Diskr. Mat., 2006, vol. 9, pp. 152–163. (in Russian)
3. *Aki S. and Hirano K.* Discrete distributions related to succession events in two-state Markov chain. Statistical Science and Data Analysis. Eds. K. Matusita, M. L. Puri, T. Hayakawa. Zeist, VSP International Science Publishers, 1993, pp. 467–474.

4. *Aki S. and Hirano K.* Sooner and later waiting time problems for runs in Markov dependent bivariate trials. Ann. Inst. Stat. Math., 1999, vol. 51, pp. 17–29.
5. *Han Q. and Aki S.* Formulae and recursions for the joint distributions of success runs of several lengths in a two-state Markov chain. Stat. Probab. Lett., 1998, vol. 40, no. 3, pp. 203–214.
6. *Savel'ev L. Ja. and Balakin S. V.* The joint distribution of the number of ones and the number of 1-runs in binary Markov sequences. Discr. Math. Appl., 2004, vol. 14, no. 4, pp. 353–372.
7. *Savel'ev L. Ja. and Balakin S. V.* A combinatorial approach to calculation of moments of characteristics of runs in ternary Markov sequences. Discr. Math. Appl., 2011, vol. 21, no. 1, pp. 47–67.
8. *Savel'ev L. Ja.* Raspredeleniya chisla sostoyaniy v dvoichnykh markovskikh stokhasticheskikh modelyakh [Calculation of the number of states in binary Markov stochastic models]. Sib. Zh. Vychisl. Mat., 2015, vol. 18, no. 2, pp. 191–200. (in Russian)
9. *Savel'ev L. Ja. and Balakin S. V.* Nekotorye primeneniya stokhasticheskoy teorii seriy [Some applications of the stochastic theory of runs]. Sib. Zh. Ind. Mat., 2012, vol. 15, no. 3, pp. 111–123. (in Russian)
10. *Shinde R. L. and Kotwal K. S.* On the joint distribution of runs in the sequence of Markov-dependent multi-state trials. Stat. Probab. Lett., 2006, vol. 76, no. 10, pp. 1065–1074.
11. *Geske M. X., Godbole A. P., Schaffner A. A., et al.* Compound Poisson approximations for word patterns under Markovian hypotheses. J. Appl. Probab., 1995, vol. 32, pp. 877–892.
12. *Erhardsson T.* Compound Poisson approximation for Markov chains using Stein's method. Ann. Probab., 1999, vol. 27, no. 1, pp. 565–596.
13. *Chryssaphinou O. and Vaggelatou E.* Compound Poisson approximation for multiple runs in a Markov chain. Ann. Inst. Stat. Math., 2002, vol. 54, no. 2, pp. 411–424.
14. *Fu J. C., Wang L., and Lou W. Y. W.* On exact and large deviation approximation for the distribution of the longest run in a sequence of two-state Markov dependent trials. J. Appl. Probab., 2003, vol. 40, no. 2, pp. 346–360.
15. *Eryilmaz S.* Some results associated with the longest run statistic in a sequence of Markov dependent trials. Appl. Math. Comput., 2006, vol. 175, no. 1, pp. 119–130.
16. *Pinsky M. A. and Karlin S.* The long run behavior of Markov chains. An Introduction to Stochastic Modeling. Fourth Edition, eds. M. A. Pinsky, S. Karlin. Boston, Elsevier, 2011, pp. 165–222.
17. *Fu J. C. and Johnson B. C.* Approximate probabilities for runs and patterns in i.i.d. and Markov-dependent multistate trials. Adv. Appl. Probab. Appl. Probab. Trust, 2009, vol. 41, no. 1, pp. 292–308.
18. *Mikhailov V. G. and Shoitov A. M.* On repetitions of long tuples in a Markov chain. Discr. Math. Appl., 2015, vol. 25, no. 5, pp. 295–303.
19. *Mikhailov V. G.* Estimates of accuracy of the Poisson approximation for the distribution of number of runs of long string repetitions in a Markov chain. Discr. Math. Appl., 2016, vol. 26, no. 2, pp. 105–113.
20. *Mytalas G. C. and Zazanis M. A.* Central Limit Theorem approximations for the number of runs in Markov-dependent binary sequences. J. Stat. Plan. Inference, 2013, vol. 143, no. 2, pp. 321–333.
21. *Mahmoudzadeh E., Montazeri M. A., Zekri M., and Sadri S.* Extended hidden Markov model for optimized segmentation of breast thermography images. Infrared Phys. Technol., 2015, vol. 72, pp. 19–28.
22. *Yang W., Tao J., and Ye Z.* Continuous sign language recognition using level building based on fast hidden Markov model. Pattern Recognit. Lett., 2016, vol. 78, pp. 28–35.

23. *Elliott R. J., Aggoun L., and Moore J. B.* Hidden Markov Models. N. Y., Springer, 1995, vol. 29, 382 p.
24. *Aston J. A. D. and Martin D. E. K.* Distributions associated with general runs and patterns in hidden Markov models. Ann. Appl. Stat., 2007, vol. 1, no. 2, pp. 585–611.
25. *Mezhennaya N. M.* O chisle sovpadeniy znakov v diskretnoy sluchaynoy posledovatel'nosti, upravlyaemoy tsep'yu Markova [On the number of characters matchings in discrete random sequence controlled by Markov chain]. Sib. Elektron. Mat. Izv., 2016, vol. 13, pp. 305–317. (in Russian)
26. *Rozanov Yu. A.* Sluchaynye protsessy. Kratkiy kurs [Random processes. Short course]. Moscow, Nauka Publ., 1979, 184 p. (in Russian)
27. *Mezhennaya N. M.* On the distribution of the number of runs in polynomial sequence controlled by Markov chain. OP&PM Surv. Appl. Ind. Math., 2016, vol. 23, no. 2, pp. 186–187.
28. *Mezhennaya N. M.* О предельном распределении числа серий в полиномиальной последовательности, управляемой цепью Маркова [On the limit distribution of a number of runs in polynomial sequence controlled by Markov chain]. Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki, 2016, vol. 26, iss. 3, pp. 324–335. (in Russian)
29. *Barbour A. D., Holst L., and Janson S.* Poisson Approximation. Oxford, Oxford Univ. Press, 1992, 277 p.

УДК 519.719.1

ОЦЕНКА РАНГА СЛУЧАЙНОЙ КВАДРАТИЧНОЙ ФОРМЫ НАД КОНЕЧНЫМ ПОЛЕМ

А. В. Черемушкин

ФГУП «НИИ «Квант», г. Москва, Россия

Изучаются свойства распределения ранга случайной квадратичной формы над конечным полем $\mathrm{GF}(q)$. Отдельно рассматриваются случаи чётной и нечётной характеристик поля. Доказаны асимптотические нижние оценки значения ранга для почти всех квадратичных форм f от n переменных вида

$$\mathrm{rank}(f) \geqslant n - \left\lceil \sqrt{2\log_q n} + c \right\rceil + 1,$$

$0 < c < 1$, при $n \to \infty$.

Ключевые слова: *конечное поле, симплектическая группа, квадратичная форма.*

DOI 10.17223/20710410/35/3

ON THE RANK OF RANDOM QUADRATIC FORM OVER FINITE FIELD

A. V. Cheremushkin

Technology Federal State Unitary Enterprise "Research Institute Kvant", Moscow, Russia

E-mail: avc238@mail.ru

The weights of Boolean functions of degree two are closely related to the ranks of quadratic forms. Though the weights of Reed — Muller codes are intensively researched in coding theory, the ranks of quadratic forms are not sufficiently studied. The article contains some asymptotic properties of the rank of a random quadratic form $f(x_1, \ldots, x_n) = \sum\limits_{1 \leqslant i,j \leqslant n} a_{ij} x_i x_j$ over finite field $\mathrm{GF}(q)$. The cases of odd and even characteristics are separately considered. If q is odd, then the $\mathrm{rank}(f)$ of f is defined as the rank of the symmetric matrix (b_{ij}) with $b_{ii} = a_{ii}$, $b_{ij} = (a_{ij} + a_{ji})/2$, $j \neq i$, $1 \leqslant i, j \leqslant n$. It is proved that, for any odd q and $0 < c < 1$, the lower bound for the rank of the almost all quadratic forms f in n variables is the following: $\mathrm{rank}(f) \geqslant n - \left\lceil \sqrt{2\log_q n} + c \right\rceil + 1$ as $n \to \infty$. In the case of even q, the $\mathrm{rank}(f)$ of f is defined as the rank of a bilinear form $f(x+y) + f(x) + f(y)$. If $q = 2^m$, $m \geqslant 1$, $n = 2k + \varepsilon$, $\varepsilon \in \{0, 1\}$, $0 < c < 1$, then the lower bound for the rank of the almost all quadratic forms f in n variables is the following: $\mathrm{rank}(f) \geqslant 2k - 2\left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c + (-1)^\varepsilon}{4} \right\rceil + 2$ as $k \to \infty$. An another form of this inequality is $\mathrm{rank}(f) > n - \left\lceil \sqrt{2\log_q n} + c' \right\rceil + 1$, where $0 < c' < 1$. As a corollary, an asymptotic lower bound for the minimal size $\beta_0(G)$ of a vertex cover of a graph G with n vertices is obtained. The vertex cover of G is a set S of vertices in G such that every arc in G is incident to a vertex in S. Let $n = 2k + \varepsilon$, $\varepsilon = 0, 1$, $c > 0$. Then for the almost all graphs G with n vertices, the lower bound for the minimal vertex cover size is the following: $\beta_0(G) \geqslant k - \left\lceil \sqrt{\frac{1}{2}\log_2 k} + \frac{c + (-1)^\varepsilon}{4} \right\rceil + 1$ as $k \to \infty$.

Keywords: *finite field, symplectic group, quadratic form.*

1. Случай конечного поля нечётного порядка

Каждой квадратичной форме

$$f(x_1,\dots,x_n)=\sum_{1\leqslant i,j\leqslant n}a_{ij}x_ix_j$$

над полем $\mathrm{GF}(q)$, $q=p^m$, $p\geqslant 3$, от $n\geqslant 1$ переменных можно поставить в соответствие симметричную матрицу (b_{ij}) с элементами $b_{ii}=a_{ii}$ и $b_{ij}=(a_{ij}+a_{ji})/2$, $j\neq i$, $1\leqslant i$, $j\leqslant n$. Ранг этой матрицы называется рангом квадратичной формы. Из классификации квадратичных форм [1, 2] следует, что квадратичную форму можно линейным преобразованием аргументов привести к одному из следующих вариантов:

(a) $x_1x_2+x_3x_4+\ldots+x_{2s-1}x_{2s}+x_{2s+1}^2$, $r=2s+1\leqslant n$;

(b) $x_1x_2+x_3x_4+\ldots+x_{2s-1}x_{2s}+dx_{2s+1}^2$, $r=2s+1\leqslant n$ и элемент $d\in\mathrm{GF}(q)^*$ не является квадратичным вычетом;

(c) $x_1x_2+x_3x_4+\ldots+x_{2s-1}x_{2s}$, $r=2s\leqslant n$;

(d) $x_1x_2+x_3x_4+\ldots+x_{2s-1}^2-dx_{2s}^2$, $r=2s\leqslant n$ и элемент $d\in\mathrm{GF}(q)^*$ не является квадратичным вычетом.

Вероятность того, что квадратичная форма от n переменных имеет ранг r, определяется как относительная доля таких форм:

$$p_r(n)=Q_r(n)/q^{n(n+1)/2},\tag{1}$$

где $Q_r(n)$ — число квадратичных форм от n переменных, имеющих ранг r; $q^{n(n+1)/2}$ — число симметричных матриц над полем $\mathrm{GF}(q)$ размера $n\times n$.

Группы инерции этих квадратичных форм имеют следующие порядки [1, 3]:

– для случаев (a) и (b)

$$2(q^{2s}-1)q^{2s-1}\dots(q^2-1)q^1=2q^{s^2}\prod_{i=1}^{s}(q^{2i}-1),$$

– для случая (c)

$$2(q^{2s-1}-q^{s-1})(q^{2s-2}-1)q^{2s-3}\dots(q^2-1)q^1,$$

– для случая (d)

$$2(q^{2s-1}+q^{s-1})(q^{2s}-2)q^{2s-3}\dots(q^2-1)q^1.$$

Соответственно для числа квадратичных форм возможны следующие значения:

$$\begin{gathered}Q_{2s+1}(n)=\frac{(q^n-1)\dots(q^n-q^{2s})}{(q^{2s}-1)q^{2s-1}\dots(q^2-1)q^1}=\\ =\frac{(q^n-1)(q^{n-1}-1)q^1(q^{n-2}-1)q^2}{(q^{2s}-1)q^{2s-1}}\cdot\frac{(q^{n-3}-1)q^3(q^{n-4}-1)q^4}{(q^{2s-2}-1)q^{2s-3}}\cdot\ldots\times\\ \times\frac{(q^{n-2s+1}-1)q^{2s-1}(q^{n-2s}-1)q^{2s}}{(q^2-1)q^1}=\\ =\frac{q^{s(s+1)}(q^n-1)(q^{n-1}-1)\cdot\ldots\cdot(q^{n-2s}-1)}{(q^{2s}-1)(q^{2s-2}-1)\cdot\ldots\cdot(q^2-1)};\end{gathered}\tag{2}$$

$$
\begin{gathered}
Q_{2s}(n) = \frac{(q^n-1)(q^n-q)\dots(q^n-q^{2s-1})}{2(q^{2s-1}-q^{s-1})(q^{2s-2}-1)q^{2s-3}\dots(q^2-1)q^1} + \\
+ \frac{(q^n-1)(q^n-q)\dots(q^n-q^{2s-1})}{2(q^{2s-1}+q^{s-1})(q^{2s-2}-1)q^{2s-3}\dots(q^2-1)q^1} = \\
= \left(\frac{(q^n-1)(q^n-q)q^{2s-1}}{(q^s-1)(q^s+1)q^{2s-2}}\right)\frac{(q^n-q^2)(q^n-q^3)\dots(q^n-q^{2s-1})}{(q^{2s-2}-1)q^{2s-3}\dots(q^2-1)q^1} = \\
= \frac{(q^n-1)(q^{n-1}-1)q^{2s+1}}{(q^{2s}-1)q^{2s-1}} \cdot \frac{(q^{n-2}-1)q^2(q^{n-3}-1)q^3}{(q^{2s-2}-1)q^{2s-3}} \cdot \dots \times \\
\times \frac{(q^{n-2s+2}-1)q^{2s-2}(q^{n-2s+1}-1)q^{2s-1}}{(q^2-1)q^1} = \frac{(q^n-1)(q^{n-1}-1)q^{2s}}{(q^{2s}-1)} \times \\
\times \frac{(q^{n-2}-1)q^2(q^{n-3}-1)}{(q^{2s-2}-1)} \cdot \dots \cdot \frac{(q^{n-2s+2}-1)q^{2s-2}(q^{n-2s+1}-1)}{(q^2-1)} = \\
= \frac{q^{s(s+1)}(q^n-1)(q^{n-1}-1)\cdot\dots\cdot(q^{n-2s+1}-1)}{(q^{2s}-1)(q^{2s-2}-1)\cdot\dots\cdot(q^2-1)}.
\end{gathered} \tag{3}
$$

Приведём свойства этих чисел, необходимые в дальнейшем.

Лемма 1. При нечётном $q \geqslant 3$ и $0 \leqslant 2s \leqslant n-2$ справедливы следующие соотношения:

$$
\frac{Q_{2s+1}(n)}{Q_{2s+3}(n)} = \frac{1}{(q^{n-2s-1}-1)(q^{n-2s-2}-1)}\left(1-\frac{1}{q^{2s+2}}\right); \tag{4}
$$

$$
\frac{Q_{2s}(n)}{Q_{2s+2}(n)} = \frac{1}{(q^{n-2s}-1)(q^{n-2s-1}-1)}\left(1-\frac{1}{q^{2s+2}}\right); \tag{5}
$$

$$
\frac{Q_{2s}(n)}{Q_{2s+1}(n)} = \frac{1}{q^{n-2s}-1}; \tag{6}
$$

$$
\frac{Q_{2s+1}(n)}{Q_{2s+2}(n)} = \frac{1}{(q^{n-2s-1}-1)}\left(1-\frac{1}{q^{2s+2}}\right). \tag{7}
$$

Поэтому числа $Q_s(n)$, $1 \leqslant s \leqslant n$, образуют монотонно возрастающие последовательности.

Доказательство. Из формул (2) и (3) вытекают соотношения

$$
\begin{gathered}
\frac{Q_{2s+3}(n)}{Q_{2s+1}(n)} = \frac{(q^{n-2s-1}-1)(q^{n-2s-2}-1)q^{2s+2}}{(q^{2s+2}-1)}, \\
\frac{Q_{2s+2}(n)}{Q_{2s}(n)} = \frac{(q^{n-2s}-1)(q^{n-2s-1}-1)q^{2s+2}}{(q^{2s+2}-1)}, \\
\frac{Q_{2s+1}(n)}{Q_{2s}(n)} = q^{n-2s}-1, \\
\frac{Q_{2s+2}(n)}{Q_{2s+1}(n)} = \frac{(q^{n-2s-1}-1)q^{2s+2}}{(q^{2s+2}-1)}.
\end{gathered}
$$

Лемма доказана. ■

Лемма 2. При нечётном $q \geqslant 3$ для числа квадратичных форм с максимальным значением ранга справедливы следующие формулы:

$$Q_{2k}(2k) = q^{k(2k+1)}\left(1-\frac{1}{q^{n-1}}\right)\left(1-\frac{1}{q^{n-3}}\right)\dots\left(1-\frac{1}{q}\right),$$
$$Q_{2k+1}(2k+1) = q^{k(2k+1)}\left(1-\frac{1}{q^{n}}\right)\left(1-\frac{1}{q^{n-2}}\right)\dots\left(1-\frac{1}{q}\right).$$

Доказательство. При $n = 2k$ имеем

$$Q_{2k}(2k) = \frac{q^{k(k+1)}(q^n-1)(q^{n-1}-1)\cdot\ldots\cdot(q^{n-2k+1}-1)}{(q^{2k}-1)(q^{2k-2}-1)\cdot\ldots\cdot(q^2-1)} =$$
$$= q^{k(k+1)}(q^{n-1}-1)(q^{n-3}-1)\dots(q^1-1) =$$
$$= q^{k(2k+1)}\left(1-\frac{1}{q^{n-1}}\right)\left(1-\frac{1}{q^{n-3}}\right)\dots\left(1-\frac{1}{q}\right).$$

Аналогично при $n = 2k+1$ имеем

$$Q_{2k+1}(2k+1) = \frac{q^{k(k+1)}(q^n-1)(q^{n-1}-1)\cdot\ldots\cdot(q^{n-2k}-1)}{(q^{2k}-1)(q^{2k-2}-1)\cdot\ldots\cdot(q^2-1)} =$$
$$= q^{k(k+1)}(q^n-1)(q^{n-2}-1)\cdot\ldots\cdot(q^1-1) =$$
$$= q^{k(2k+1)}\left(1-\frac{1}{q^{n}}\right)\left(1-\frac{1}{q^{n-2}}\right)\dots\left(1-\frac{1}{q}\right).$$

Лемма доказана. ■

Лемма 3. При нечётном $q \geqslant 3$ для вероятности максимального значения ранга справедливы следующие формулы:

$$p_{2k}(2k) = \left(1-\frac{1}{q^{2k-1}}\right)\left(1-\frac{1}{q^{2k-3}}\right)\dots\left(1-\frac{1}{q}\right),$$
$$p_{2k+1}(2k+1) = \left(1-\frac{1}{q^{2k+1}}\right)\left(1-\frac{1}{q^{2k-1}}\right)\dots\left(1-\frac{1}{q}\right).$$

В частности, $p_{2k+1}(2k+1) = \left(1-\dfrac{1}{q^{2k+1}}\right)p_{2k}(2k)$ при $k \geqslant 2$.

Доказательство непосредственно следует из формулы (1) и леммы 2.

Лемма 4. При нечётном $q \geqslant 3$, $n \geqslant 2$ и $1 \leqslant i \leqslant n$ справедлива оценка

$$\frac{Q_{n-i}(n)}{Q_n(n)} < \frac{1}{(q^i-1)(q^{i-1}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)}.$$

Доказательство. Из равенств (4)–(7) леммы 1 следует, что при $n = 2k$ и $1 \leqslant i \leqslant k$ выполнены равенства

$$\frac{Q_{2k-2i}(2k)}{Q_{2k}(2k)} < \frac{1}{(q^{2i}-1)(q^{2i-1}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)},$$
$$\frac{Q_{2k-2i-1}(2k)}{Q_{2k}(2k)} < \frac{1}{(q^{2i+1}-1)(q^{2i}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)}.$$

Таким образом, при всех $1 \leqslant i \leqslant 2k$ справедлива оценка

$$\frac{Q_{2k-i}(2k)}{Q_{2k}(2k)} < \frac{1}{(q^i-1)(q^{i-1}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)}.$$

При нечётном $n = 2k+1$ и $1 \leqslant i \leqslant k$ аналогично получаем

$$\frac{Q_{2k-2i+1}(2k+1)}{Q_{2k+1}(2k+1)} < \frac{1}{(q^{2i}-1)(q^{2i-1}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)},$$
$$\frac{Q_{2k-2i}(2k+1)}{Q_{2k+1}(2k+1)} < \frac{1}{(q^{2i+1}-1)(q^{2i}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)}.$$

Таким образом, при всех $1 \leqslant i \leqslant 2k+1$ справедлива оценка

$$\frac{Q_{2k+1-i}(2k+1)}{Q_{2k+1}(2k+1)} < \frac{1}{(q^i-1)(q^{i-1}-1)\cdot\ldots\cdot(q^2-1)(q^1-1)}.$$

Лемма доказана. ■

Теорема 1. Пусть q нечётно и $0 < c < 1$. При $n \to \infty$ для ранга почти всех квадратичных форм f от n переменных при $n \to \infty$ справедлива оценка

$$\operatorname{rank}(f) \geqslant n - \left\lceil \sqrt{2\log_q n} + c \right\rceil + 1.$$

Доказательство. Оценим долю квадратичных форм от n переменных ранга, не превышающего $n-i$:

$$\sum_{j=0}^{n-i} Q_j(n) q^{-n(n+1)/2} < \sum_{j=0}^{n-i} \frac{Q_j(n)}{Q_n(n)} < n\frac{Q_{n-i)}(n)}{Q_n(n)} < \frac{n}{(q^1-1)(q^2-1)\cdot\ldots\cdot(q^{i-1}-1)(q^i-1)}$$

(здесь использована оценка из леммы 4). В результате имеем

$$\log_q\left(\sum_{j=0}^{n-i}\frac{Q_j(n)}{Q_n(n)}\right) < \log_q n - \sum_{m=1}^{i}\log_q(q^m-1) = \log_q n - \sum_{m=1}^{i}\left(m - \log_q\left(1-\frac{1}{q^m}\right)\right) =$$
$$= \log_q n - \frac{i(i+1)}{2} + \sum_{m=1}^{i}\log_q\left(1-\frac{1}{q^m}\right) < \log_q n - \frac{i(i+1)}{2}.$$

При $i = \left\lceil \sqrt{2\log_q n} + c \right\rceil$, $c > 0$, выражение

$$\log_q n - \frac{i(i+1)}{2} < \log_q n - \frac{i^2}{2} \leqslant \log_q n - \frac{\left(\sqrt{2\log_q k} + c\right)^2}{2} = -c\sqrt{2\log_q n} + \frac{c^2}{2}$$

стремится к $-\infty$ при $n \to \infty$. Поэтому при $n \to \infty$ доля квадратичных форм ранга, меньшего чем $n - \left\lceil \sqrt{2\log_q n} + c \right\rceil$, стремится к нулю. ■

2. Случай конечного поля чётного порядка

Каждая квадратичная форма f над полем $\mathrm{GF}(q)$, $q = 2^m$, $m \geqslant 1$, характеристики два от n переменных может иметь только чётный ранг, причем форму ранга $2r$, $1 \leqslant r \leqslant \lfloor n/2 \rfloor$, можно линейным преобразованием аргументов привести к виду

а) $x_1x_2 + x_3x_4 + \ldots + x_{2r-1}x_{2r} + x_{2r+1}^2$ $(2r+1 \leqslant n)$,

б) $x_1x_2 + x_3x_4 + \ldots + x_{2r-1}x_{2r} + \alpha x_{2r-1}^2 + \alpha x_{2r}^2$ $(2r \leqslant n)$,

где $\alpha = 0$ либо α является корнем неприводимого над $\mathrm{GF}(q)$ многочлена $\alpha x^2 + x + \alpha$ [1]. Так как ранг определяется по билинейной знакопеременной форме

$$f(x+y) + f(x) + f(y),$$

$x, y \in \mathrm{GF}(q)^n$, которая однозначно строится по квадратичной части многочлена, содержащей смешанные произведения без квадратов, то вероятность того, что квадратичная форма от n переменных имеет ранг $2r$, можно записать в виде

$$p_{2r}(n) = Q_{2r}(n)/q^{n(n-1)/2},$$

где, как и выше, $Q_{2r}(n)$ — число квадратичных форм от n переменных, имеющих ранг $2r$, а $n(n-1)/2$ — число попарных произведений переменных. Учитывая, что группа инерции билинейной формы от $2r$ переменных ранга $2r$ совпадает с симплектической группой $\mathbf{Sp}(2r, q)$ (см., например, [1, 3, 4]), получаем

$$Q_{2r}(n) = \frac{(q^n - 1)\dots(q^n - q^{2r-1})}{|\mathbf{Sp}(2r, q)|}.$$

Так как [5, с. 414] $|\mathbf{Sp}(2r, q)| = q^{r^2} \prod\limits_{i=1}^{r} (q^{2i} - 1) = (q^{2r} - 1)q^{2r-1} \dots (q^2 - 1)q^1$, то

$$\begin{aligned} Q_{2r}(n) &= \frac{(q^n - 1)(q^n - q^1)}{(q^{2r} - 1)q^{2r-1}} \dots \frac{(q^n - q^{2r-2})(q^n - q^{2r-1})}{(q^2 - 1)q^1} = \\ &= \frac{(q^n - 1)(q^{n-1} - 1)q^0}{(q^{2r} - 1)} \dots \frac{(q^{n-2r+2} - 1)(q^{n-2r+1} - 1)q^{2r-2}}{(q^2 - 1)}. \end{aligned} \tag{8}$$

Лемма 5. При $q = 2^m$, $m \geqslant 1$, $1 \leqslant r \leqslant n/2 - 1$ справедливо соотношение

$$\frac{Q_{2r}(n)}{Q_{2r+2}(n)} = \frac{q^2}{(q^{n-2r} - 1)(q^{n-2r-1} - 1)} \left(1 - \frac{1}{q^{2r+2}}\right). \tag{9}$$

Числа $Q_{2r}(n)$, $1 \leqslant r \leqslant n/2 - 1$, образуют монотонно возрастающую последовательность.

Доказательство вытекает из равенства (8) и соотношения

$$Q_{2r+2}(n)/Q_{2r}(n) = \frac{(q^{n-2r} - 1)(q^{n-2r-1} - 1)q^{2r}}{(q^{2r+2} - 1)}.$$

Лемма 6. При $q = 2^m$, $m \geqslant 1$, для числа квадратичных форм с максимальным значением ранга справедливы следующие формулы:

$$\begin{aligned} Q_{2k}(2k) &= q^{k(2k-1)} \left(1 - \frac{1}{q^{n-1}}\right)\left(1 - \frac{1}{q^{n-3}}\right) \dots \left(1 - \frac{1}{q}\right), \\ Q_{2k}(2k+1) &= q^{k(2k+1)} \left(1 - \frac{1}{q^n}\right)\left(1 - \frac{1}{q^{n-2}}\right) \dots \left(1 - \frac{1}{q^3}\right). \end{aligned}$$

Доказательство. При $n = 2k$ имеем

$$\begin{aligned} Q_{2k}(2k) &= \frac{(q^n - 1)(q^{n-1} - 1)q^0}{(q^{2k} - 1)} \dots \frac{(q^2 - 1)(q^1 - 1)q^{2k-2}}{(q^2 - 1)} = \\ &= q^{k(k-1)}(q^{n-1} - 1)(q^{n-3} - 1)\dots(q^1 - 1) = q^{k(2k-1)} \left(1 - \frac{1}{q^{n-1}}\right)\left(1 - \frac{1}{q^{n-3}}\right) \dots \left(1 - \frac{1}{q}\right). \end{aligned}$$

Аналогично при $n = 2k + 1$ имеем

$$Q_{2k}(2k+1) = q^{k(2k+1)} \left(1 - \frac{1}{q^n}\right) \left(1 - \frac{1}{q^{n-2}}\right) \dots \left(1 - \frac{1}{q^3}\right).$$

Лемма доказана. ■

Лемма 7. При $q = 2^m$, $m \geqslant 1$, для вероятности максимального значения ранга справедливы следующие формулы:

$$p_{2k}(2k) = \left(1 - \frac{1}{q^{2k-1}}\right) \left(1 - \frac{1}{q^{2k-3}}\right) \dots \left(1 - \frac{1}{q}\right),$$
$$p_{2k}(2k+1) = \left(1 - \frac{1}{q^{2k+1}}\right) \left(1 - \frac{1}{q^{2k-1}}\right) \dots \left(1 - \frac{1}{q^3}\right).$$

В частности, $p_{2k}(2k) = \left(1 - \dfrac{1}{q}\right) p_{2k-2}(2k-1)$ при $k \geqslant 1$.

Доказательство непосредственно следует из леммы 6.

Теорема 2. Пусть $q = 2^m$, $m \geqslant 1$, $n = 2k+\varepsilon$, $\varepsilon \in \{0, 1\}$ и $0 < c < 1$ — произвольная константа. При $n \to \infty$ для ранга почти всех квадратичных форм f от n переменных при $n \to \infty$ справедлива оценка

$$2k \geqslant \operatorname{rank}(f) \geqslant 2k - 2 \left\lceil \sqrt{\frac{1}{2} \log_q k} + \frac{c + (-1)^\varepsilon}{4} \right\rceil + 2.$$

Доказательство. В силу равенства (9) имеем

$$Q_{2r}(n) < \frac{q^2}{(q^{n-2r-1} - 1)(q^{n-2r} - 1)} Q_{2r+1}(n), \tag{10}$$

поэтому при $n = 2k$ и $1 \leqslant i \leqslant k$ справедлива оценка

$$\frac{Q_{2(k-i)}(2k)}{Q_{2k}(2k)} < \frac{q^{2i}}{(q^2-1)(q^3-1)(q^4-1) \cdot \ldots \cdot (q^{n-2(k-i)-1} - 1)(q^{n-2(k-i)} - 1)} =$$
$$= \frac{q^{2i}}{(q^2-1)(q^3-1)(q^4-1) \cdot \ldots \cdot (q^{2i-1} - 1)(q^{2i} - 1)}.$$

Оценим долю квадратичных форм от n переменных ранга, не превышающего $2k - 2i$:

$$\sum_{j=0}^{k-i} Q_{2j}(2k) q^{-\frac{n(n-1)}{2}} < \sum_{j=0}^{k-i} \frac{Q_{2j}(2k)}{Q_{2k}(2k)} < k \frac{Q_{2(k-i)}(2k)}{Q_{2k}(2k)} <$$
$$< \frac{kq^{2i}}{(q^2-1)(q^3-1)(q^4-1) \cdot \ldots \cdot (q^{2i-1} - 1)(q^{2i} - 1)}.$$

Оценим логарифм этого выражения:

$$\log_q \left(\sum_{j=0}^{k-i} \frac{Q_{2j}(2k)}{Q_{2k}(2k)} \right) < \log_q k + 2i - \sum_{m=2}^{2i} \log_q (q^m - 1) =$$
$$= \log_q k + 2i - \sum_{m=2}^{2i} \left(m - \log_q \left(1 - \frac{1}{q^m}\right) \right) =$$
$$= \log_q k + 2i - \frac{(2i+2)(2i-1)}{2} + \sum_{m=2}^{2i} \log_q \left(1 - \frac{1}{q^m}\right) <$$

$$< \log_q k - 2i^2 + i + 1. \qquad (11)$$

Положим $i = \left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c+1}{4} \right\rceil$, где $c > 0$. При таком выборе при $k \to \infty$ выражение (11) стремится к $-\infty$:

$$\log_q k - 2i^2 + i + 1 \leqslant \log_q k - 2\left(\sqrt{\frac{1}{2}\log_q k} + \frac{c+1}{4}\right)^2 + \left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c+1}{4} \right\rceil + 1 <$$

$$< \log_q k - 2\left(\frac{1}{2}\log_q k + 2\sqrt{\frac{1}{2}\log_2 k}\frac{c+1}{4} + \left(\frac{c+1}{4}\right)^2\right) + \sqrt{\frac{1}{2}\log_q k} + \frac{c+1}{4} + 2 =$$

$$= -c\sqrt{\frac{1}{2}\log_q k} + \frac{2c^2 + c + 1}{4} + 2.$$

Поэтому для $n = 2k$ при $k \to \infty$ доля квадратичных форм ранга, меньшего чем $2k - \left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c+1}{4} \right\rceil$, стремится к нулю.

При $n = 2k+1$ и $1 \leqslant i \leqslant k$ рассуждаем аналогично. В силу неравенства (10) справедлива оценка

$$\frac{Q_{2(k-i)}(2k+1)}{Q_{2k}(2k+1)} < \frac{q^{2i}}{(q^2-1)(q^3-1)\cdot\ldots\cdot(q^{n-2(k-i)-1}-1)(q^{n-2(k-i)}-1)} =$$

$$= \frac{q^{2i}}{(q^2-1)(q^3-1)\cdot\ldots\cdot(q^{2i}-1)(q^{2i+1}-1)}.$$

Аналогично получаем

$$\sum_{j=0}^{k-i} Q_{2j}(2k+1)q^{-\frac{n(n-1)}{2}} < \sum_{j=0}^{k-i}\frac{Q_{2j}(2k+1)}{Q_{2k}(2k+1)} < k\frac{Q_{2(k-i)}(2k+1)}{Q_{2k}(2k+1)} <$$

$$< \frac{kq^{2i}}{(q^2-1)(q^3-1)\cdot\ldots\cdot(q^{2i}-1)(q^{2i+1}-1)}.$$

Оценим логарифм этого выражения:

$$\log_q\left(\sum_{j=0}^{k-i}\frac{Q_{2j}(2k+1)}{Q_{2k}(2k+1)}\right) < \log_q k - \sum_{m=2}^{2i+1}\log_q(q^m-1) =$$

$$= \log_q k + 2i - \sum_{m=2}^{2i+1}\left(i - \log_q\left(1 - \frac{1}{q^i}\right)\right) < \log_q k + 2i - \tfrac{(2i+3)2i}{2} = \log_q k - 2i^2 - i.$$

Полагая $i = \left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c-1}{4} \right\rceil$, получаем, что при $k \to \infty$ последнее выражение стремится к $-\infty$. Поэтому для $n = 2k+1$ при $k \to \infty$ доля квадратичных форм ранга, меньшего чем $2k - 2\left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c-1}{4} \right\rceil$, стремится к нулю. ■

Можно изменить вид нижней оценки из теоремы 2, приблизив его к виду оценки из теоремы 1.

Следствие 1. Пусть $q = 2^m$, $m \geqslant 1$, $k = \lfloor n/2 \rfloor$ и $0 < c' < 1$ — произвольная константа. При $n \to \infty$ для ранга почти всех квадратичных форм f от n переменных при $n \to \infty$ справедлива оценка $\operatorname{rank}(f) > n - \left\lceil \sqrt{2\log_q n} + c' \right\rceil + 1$.

Действительно,

$$2k - 2\left\lceil \sqrt{\frac{1}{2}\log_q k} + \frac{c + (-1)^\varepsilon}{4} \right\rceil + 2 \geqslant n - \left\lceil \sqrt{2\log_q k} + \frac{c+1}{2} \right\rceil + 1 =$$

$$= n - \left\lceil \sqrt{2\log_q n - 2\log_q \frac{n}{k}} + \frac{c+1}{2} \right\rceil + 1 > n - \left\lceil \sqrt{2\log_q n} + \frac{c+1}{2} \right\rceil + 1.$$

Приведём еще одно следствие из теоремы 2, содержащее нетривиальную асимптотическую нижнюю оценку наименьшего вершинного покрытия случайного простого, т. е. неориентированного без петель, графа. Хотя оценка представляется достаточно грубой, тем не менее автору не известно более точных результатов.

Следствие 2. Пусть $n = 2k + \varepsilon$, $\varepsilon \in \{0, 1\}$, и $c > 0$. Для почти всех простых графов с n вершинами величина наименьшего вершинного покрытия при $n \to \infty$ оценивается снизу величиной $\beta_0(G) \geqslant k - \left\lceil \sqrt{\frac{1}{2}\log_2 k} + \frac{c+(-1)^\varepsilon}{4} \right\rceil + 1$.

Доказательство. Каждому простому графу с n вершинами взаимно однозначно соответствует квадратичная форма от n переменных над полем из двух элементов, где каждому ребру (i, j) в графе соответствует одночлен $x_i x_j$, $1 \leqslant i < j \leqslant n$. При этом наименьшему вершинному покрытию графа соответствует такой набор переменных, фиксация которого произвольными значениями превращает квадратичную форму в функцию меньшей степени нелинейности. Так как наименьшее число одночленов второй степени у квадратичной формы ранга $r = 2s$ будет в её каноническом представлении (a)–(d), то число таких фиксаций ограничено снизу значением s, т. е. $\beta_0(G) \geqslant s$. Остаётся воспользоваться оценкой из теоремы 2. ■

ЛИТЕРАТУРА

1. *Dixon L. E.* Linear Groups with an Expositions to the Galois Field Theory. Leipzig: Publ. by B. G. Teubner, 1901.
2. *Mullen G. L. and Panario D.* Handbook of Finite Fields. CRC Press, A Chapman & Hall Book, 2013.
3. *Дьедонне Ж.* Геометрия классических групп. М.: Мир, 1974.
4. *Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979.
5. *Бурбаки Н.* Алгебра: модули, кольца, формы. М.: Наука, 1960.

REFERENCES

1. *Dixon L. E.* Linear Groups with an Expositions to the Galois Field Theory. Leipzig, Publ. by B. G. Teubner, 1901.
2. *Mullen G. L. and Panario D.* Handbook of Finite Fields. CRC Press, A Chapman & Hall Book, 2013.
3. *D'edonne Zh.* Geometriya klassicheskikh grupp [Geometry of Classical Groups]. Moscow, Mir Publ., 1974. (in Russian)
4. *MacWilliams F. J. and Sloane N. J. A.* The Theory of Error-Correcting Codes. North Holland, 1977.
5. *Burbaki N.* Algebra: moduli, kol'tsa, formy [Algebra: Modules, Rings Forms]. Moscow, Nauka Publ., 1960. (in Russian)

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7

О ДВУХКАСКАДНЫХ КОНЕЧНО-АВТОМАТНЫХ КРИПТОГРАФИЧЕСКИХ ГЕНЕРАТОРАХ И МЕТОДАХ ИХ КРИПТОАНАЛИЗА[1]

Г. П. Агибалов, И. А. Панкратова

Национальный исследовательский Томский государственный университет, г. Томск, Россия

Рассматривается криптографический генератор $G = A_1 \cdot A_2$, представляющий собой последовательное соединение двух абстрактных конечных автоматов A_1 и A_2 над полем $\mathbb{F}_2$ с множествами состояний $\mathbb{F}_2^n$, $n > 1$, и $\mathbb{F}_2^m$, $m > 1$, соответственно, с выходным алфавитом $\mathbb{F}_2$ и с функциями выходов $f_1(x)$ и $f_2(u, y)$ из некоторых классов булевых функций от n и $m + 1$ переменных соответственно. Автомат A_1 автономный с произвольной функцией переходов $g_1(x)$, автомат A_2 неавтономный с входным алфавитом $\mathbb{F}_2$ и функцией переходов $g_2(u, y)$, в которой $g_2(0, y) = g^\delta(y)$ и $g_2(1, y) = g^\tau(y)$ для некоторых различных нетрицательных целых δ и τ и отображения $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$. В каждый момент времени $t = 1, 2, \ldots$ автомат A_1 из состояния $x(t)$ переходит в состояние $x(t + 1) = g_1(x(t))$ и вырабатывает выходной символ $u(t) = f_1(x(t))$, автомат A_2 из состояния $y(t)$ переходит в состояние $y(t + 1) = g_2(u(t), y(t))$ и вырабатывает выходной символ $z(t) = f_2(u(t), y(t))$, который и является выходным символом генератора G. Ключом генератора может быть любой непустой набор элементов из ряда $x(1), y(1), f_1, g_1, f_2, g_2, g, \delta, \tau$. Задача криптоанализа генератора G состоит в определении его ключа по заданному конечному отрезку $\gamma = z(1)z(2)\ldots z(l)$ его выходной последовательности. Показано, что в генераторе G с линейным автоматом A_2 ключ $y(1)$ вскрывается с полиномиальной сложностью решением системы линейных уравнений, а ключ $(x(1), y(1))$ — линеаризационной атакой сложности не более 2^n. Предложен метод, позволяющий в произвольном генераторе G с известными функциями g_2 и f_2 вычислить по γ отрезок управляющей последовательности $\beta = u(1)u(2)\ldots u(l)$ на выходе A_1 и тем самым открыть две возможности для криптоанализа такого G: 1) найти его ключ $(x(1), y(1))$ атакой «встреча посредине» со сложностью 2^m и 2) свести задачу криптоанализа G к криптоанализу автомата A_1 — найти ключ последнего по β. Сложность метода полиномиальная, если $y(1)$ не входит в ключ, и не превосходит 2^m в противном случае. Если ключом в A_1 служит функция f_1, то его вскрытие, в свою очередь, сводится к доопределению частичной булевой функции со значениями $u(t)$ на состояниях $x(t)$ для $t = 1, 2, \ldots, l$ до функции в классе функции f_1. Аналогично, к доопределению частичной булевой функции со значениями $z(t)$ на парах $(u(t), y(t))$ для $t = 1, 2, \ldots, l$ до функции в классе функции f_2 сводится вскрытие ключа произвольного генератора G с ключом f_2. Сообщается об известных авторам алгоритмах доопределения частичной булевой функции от сколь угодно большого набора переменных до функции из класса полностью определённых булевых функций, существенно зависящих от малого или ограниченного количества переменных из этого набора.

[1]Работа поддержана грантом РФФИ, проект № 17-01-00354.

Ключевые слова: *конечный автомат, криптографический генератор, генератор (δ, τ)-шагов, криптоанализ, линеаризационная атака, атака «разделяй и решай», атака «разделяй — решай — подставляй», атака «встреча посредине».*

DOI 10.17223/20710410/35/4

ABOUT 2-CASCADE FINITE AUTOMATA CRYPTOGRAPHIC GENERATORS AND THEIR CRYPTANALYSIS

G. P. Agibalov, I. A. Pankratova

National Research Tomsk State University, Tomsk, Russia

E-mail: agibalov@isc.tsu.ru, pank@isc.tsu.ru

A cryptographic generator under consideration is a serial connectiom $G = A_1 \cdot A_2$ of two finite state machines (finite automata) A_1 and A_2 both defined over the two-element field $\mathbb{F}_2$. The first one is an autonomous automaton $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ with the state set $\mathbb{F}_2^n$, $n > 1$, the output alphabet $\mathbb{F}_2$, a transition function $g_1 : \mathbb{F}_2^n \to \mathbb{F}_2^n$, and an output function $f_1 : \mathbb{F}_2^n \to \mathbb{F}_2$. The second one is a non-autonomous automaton $A_2 = (\mathbb{F}_2, \mathbb{F}_2^n, \mathbb{F}_2, g_2, f_2)$ with the state set $\mathbb{F}_2^m$, $m > 1$, the input and output alphabets $\mathbb{F}_2$, an output function $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^n \to \mathbb{F}_2$, and a transition function $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2^m$, for which there exist a map $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$ and some different integers δ and τ such that, for all $u \in \mathbb{F}_2$ and $y \in \mathbb{F}_2^m$, $g_2(u, y) = \neg u g^\delta(y) + u g^\tau(y)$, where $g^0(y) = y$ and $g^r(y) = g(g^{r-1}(y))$ for every integer $r \geqslant 1$. Thus, the generator G is a finite autonomous automaton $G = A_1 \cdot A_2 = (\mathbb{F}_2^n \times \mathbb{F}_2^m, \mathbb{F}_2, h, f)$, where $h : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n \times \mathbb{F}_2^m$ and $h(x, y) = (g_1(x), g_2(f_1(x), y))$, $f : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2$ and $f(x, y) = f_2(f_1(x), y)$, $x \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2^m$. As we see, all the functions in the definition of G are Boolean ones, and the functions g_1 and g_2, g are vector functions of dimensions n and m respectively. Further, it is assumed that each of them belongs to a predetermined class of Boolean functions and the classes of functions f_1 and f_2 are denoted by C_1 and C_2 respectively. At any time moment $t = 1, 2, \ldots$, the automaton A_1 goes from its state $x(t)$ to a state $x(t + 1) = g_1(x(t))$ and produces an output symbol $u(t) = f_1(x(t))$, the automaton A_2 receives $u(t)$ and goes from its state $y(t)$ to a state $y(t + 1) = g_2(u(t), y(t))$ and produces an output symbol $z(t) = f_2(u(t), y(t))$ which is the output symbol generated by G at this moment. In general, a key of the generator can be defined as any non-empty subset of the set $\{x(1), y(1), f_1, g_1, f_2, g_2, g, \delta, \tau\}$. The cryptanalysis problem for G is the following: given a finite beginning $\gamma = z(1)z(2)\ldots z(l)$ of a sequence generated by G, find the generator key value. For solving the problem, the generator is described by a system of logical equations, connecting bits in γ with the initial states and values of transition and output functions in automata A_1 and A_2. It is shown that in G with the linear A_2, the key $y(1)$ is determined with the polynomial complexity by solving a system of linear equations and the key $(x(1), y(1))$ — by the linearization attack (trying different initial states of A_1) with the complexity 2^n which is much less than 2^{n+m} — the complexity of the bruitforce attack. For an arbitrary G with the known g_2 and f_2, a method is proposed allowing to compute from γ a string of the output sequence $\beta = u(1)u(2)\ldots u(l)$ of A_1 and so to reveal two possibilities for cryptanalysis of this G: 1) to determine its key $(x(1), y(1))$ by the meet-in-the-middle attack with the complexity 2^m or 2) to reduce the cryptanalysis problem for G to the cryptanalysis problem for A_1, that is, to determine the key of A_1 by β. The complexity of the method is polynomial if $y(1)$ is not included in the key and is not more than 2^m

otherwise. In case when the key of an arbitrary G is f_1, this key can be determined by identifying f_1 with a function $f \in C_1$ satisfying the equalities $f(x(t)) = u(t)$, $t = 1, 2, \ldots, l$. Similarly, if the key of G is f_2, the determination of f_2 is reduced to the construction of a function $f \in C_2$ satisfying the equalities $f(u(t), y(t)) = z(t)$, $t = 1, 2, \ldots, l$. In connection with the last, it is told about some algorithms, known to the authors, for extending a partially defined Boolean function in a large set of variables to a function from a class of completely defined Boolean functions essentially depending on a little or bounded number of variables in this set.

Keywords: *finite automaton, cryptographic generator, (δ, τ)-step generator, cryptanalysis, linearization attack, "devide and solve" attack, "devide — solve — substitute" attack, "meet-in-the middle" attack.*

1. Определение генератора

Рассматриваемый здесь криптографический генератор (будем обозначать его G) представляет собой последовательное соединение двух абстрактных конечных автоматов над полем $\mathbb{F}_2$ — управляющего и управляемого. Первый является некоторым автономным автоматом $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ с множеством состояний $\mathbb{F}_2^n$, $n > 1$, и выходным алфавитом $\mathbb{F}_2$, с функцией переходов $g_1 : \mathbb{F}_2^n \to \mathbb{F}_2^n$ и функцией выходов $f_1 : \mathbb{F}_2^n \to \mathbb{F}_2$, а второй — некоторым неавтономным автоматом $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$ с входным и выходным алфавитами $\mathbb{F}_2$, с множеством состояний $\mathbb{F}_2^m$, $m > 1$, с функцией выходов $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2$ и функцией переходов $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2^m$, для которой существуют отображение $g : \mathbb{F}_2^m \to \mathbb{F}_2^m$ и различные целые неотрицательные числа δ и τ, такие, что для всех $u \in \mathbb{F}_2$ и $y \in \mathbb{F}_2^m$ если $u = 0$, то $g_2(u, y) = g^\delta(y)$, и если $u = 1$, то $g_2(u, y) = g^\tau(y)$, т. е. $g_2(u, y) = \neg u g^\delta(y) + u g^\tau(y)$, где, как обычно, $g^0(y) = y$ и $g^r(y) = g(g^{r-1}(y))$ для любого целого $r \geqslant 1$. Таким образом, генератор G — это конечный автономный автомат $G = A_1 \cdot A_2 = (\mathbb{F}_2^n \times \mathbb{F}_2^m, \mathbb{F}_2, h, f)$, где $h : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n \times \mathbb{F}_2^m$ и $h(x, y) = (g_1(x), g_2(f_1(x), y))$, $f : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2$ и $f(x, y) = f_2(f_1(x), y)$, $x \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2^m$. Все функции в определении G, как видим, булевы, причём функции g_1 и g_2, g векторные размерности n и m соответственно.

Генератор G функционирует в дискретном времени $t = 1, 2, \ldots$, в каждый момент t которого его автомат A_1, находясь в состоянии $x(t) = x_1(t)x_2(t)\ldots x_n(t) \in \mathbb{F}_2^n$, выдаёт выходной управляющий символ $u(t) = f_1(x(t))$ и переходит в следующее состояние $x(t + 1) = g_1(x(t))$, а автомат A_2 в этот момент, находясь в состоянии $y(t) = y_1(t)y_2(t)\ldots y_n(t) \in \mathbb{F}_2^m$, принимает от A_1 символ $u(t)$, выдаёт свой выходной символ $v(t) = f_2(u(t), y(t))$ и переходит в следующее состояние $y(t+1) = g_2(u(t), y(t))$. Значением $z(t)$ на выходе генератора G в момент t является значение $v(t)$ на выходе автомата A_2 в этот момент.

Предполагается, что значение любой функции в генераторе G на любом наборе значений её аргументов вычисляется за полиномиальное время от числа последних.

Можно показать, что в частном случае, когда функции g_1 и g_2 являются функциями переходов некоторых регистров сдвига с линейной обратной связью и длиной n и m соответственно и $f_2(u, y_1, y_2, \ldots, y_m) = y_1$, генератор G функционально эквивалентен генератору (δ, τ)-шагов [1], в котором функция переходов второго регистра есть g.

Функционирование генератора G во времени описывается формально следующей системой E векторных булевых уравнений с переменными $x(t), y(t), u(t)$, $t = 1, 2, \ldots$:

$$\begin{cases} u(t) = f_1(x(t)), \\ x(t+1) = g_1(x(t)), \\ x(1) = x_1(1)x_2(1)\dots x_n(1); \\ z(t) = f_2(u(t), y(t)), \\ y(t+1) = g_2(u(t), y(t)) = \neg u(t) g^{\delta}(y(t)) + u(t) g^{\tau}(y(t)), \\ y(1) = y_1(1)y_2(1)\dots y_m(1); \\ t \geqslant 1. \end{cases}$$

В ней подсистема E_1 из первого, второго и третьего уравнений описывает работу управляющего автомата A_1, подсистема E_2 из четвёртого, пятого и шестого уравнений — работу управляемого автомата A_2.

Ключом генератора G теоретически может быть любое непустое подмножество множества $\{x(1), y(1), f_1, g_1, f_2, g_2, g, \delta, \tau\}$. Требование стойкости криптографического генератора накладывает определённые ограничения на применяемые в нём булевы функции. Кроме того, не любую булеву функцию можно задать практически. В этой связи предполагается, что каждая функция в генераторе принадлежит некоторому классу функций, ограниченных по сложности и обладающих некоторыми криптографическими свойствами, и этот класс, в том числе для функции в составе ключа, общеизвестен. При известном ключе и заданных других параметрах генератора уравнения данной системы E позволяют однозначно вычислить порождаемую генератором выходную последовательность $z(1)z(2)\dots$

Здесь мы ограничимся случаями, в которых ключ генератора есть $x(1)$ — начальное состояние A_1, $y(1)$ — начальное состояние A_2, $(x(1), y(1))$ — начальное состояние G, f_1 — функция выходов A_1 или f_2 — функция выходов A_2. Далее классы функций f_1 и f_2 обозначаются C_1 и C_2 соответственно.

2. Постановка задачи

Задача криптоанализа произвольного генератора G заключается в определении его ключа в известном классе по заданному конечному отрезку $\gamma = z(1)z(2)\dots z(l)$ последовательности, порождаемой им на этом ключе. В такой постановке задача возникает в криптоанализе поточного шифра, использующего данный генератор, атакой на шифр с известным или выбираемым открытым текстом. Её решение может быть получено как решение конечной подсистемы $E(l)$ указанной системы уравнений E с $t = 1, 2, \dots, l$. В случае неединственности решения системы $E(l)$ обычно рекомендуется задаться более длинным отрезком γ.

Решение системы $E(l)$ может быть найдено атакой грубой силы, или исчерпывающим поиском, т. е. перебором возможных ключей с вычислением при каждом выбранном ключе k начального отрезка γ' длины l порождаемой последовательности и сравнением его с отрезком γ. В случае $\gamma' = \gamma$ ключ k принимается за ответ задачи. Сложность этой атаки определяется размером ключевого пространства. Так, если ключом генератора служит его начальное состояние, то сложность атаки равна 2^{n+m}. Если же ключом является какая-либо из функций генератора, то сложность атаки равна мощности класса этой функции. Ниже приведены примеры атак на любой генератор G со сложностью меньше, чем у атаки грубой силы. О некоторых из этих атак докладывалось на 15 Сибирской школе-семинаре «Компьютерная безопасность и криптография» [2].

3. Атака на состояние генератора

Подмножество L (скалярных) переменных в некоторой системе уравнений S над некоторым кольцом называется *эффективным множеством*, если фиксирование любых возможных значений этих переменных превращает S в легко решаемую (например, за полиномиальное или меньшее время) систему уравнений (ЛРС). В частности, таковым является подмножество переменных, при любом фиксировании которых все уравнения в системе превращаются в линейные. Оно называется линеаризационным множеством переменных системы [3, 4]. Очевидно, что для любой ЛРС пустое множество переменных эффективное, а в системе булевых уравнений (СБУ) с r переменными любые $r-1$ переменных образуют линеаризационное множество.

Всякая СБУ S с q-элементным эффективным множеством L решается со сложностью 2^q методом DS (Devide and Solve), или по-русски «разделяй и решай», состоящим в подстановке в систему S поочерёдно различных наборов значений переменных в L и в решении получаемой каждый раз ЛРС. В случае совместности последней её решение, взятое вместе с подставленным в S набором значений переменных в L, является решением системы S. Метод DS на основе линеаризационного множества переменных называется линеаризационной атакой [3, 4]. В нём решается частный случай ЛРС — система линейных уравнений (СЛУ).

Автомат A_2 называется линейным, если все координатные функции в g_2 и функция f_2 линейные. Нетрудно заметить, что в системе $E(l)$ уравнений генератора G с линейным управляемым автоматом A_2 и ключом $(x(1), y(1))$ переменные $x_1(1)$, $x_2(1)$, …, $x_n(1)$ в $x(1)$ образуют линеаризационное множество системы уравнений $E(l)$, ибо фиксирование криптоаналитиком определённых значений этих переменных позволяет ему вычислить и тем самым зафиксировать в $E(l)$ определённые значения остальных переменных, кроме $y_j(t)$ для $j = 1, 2, \ldots, m$ и $t = 1, 2, \ldots, l$, и, благодаря линейности функций g_2 и f_2, превратить $E(l)$ в некоторую СЛУ с переменными из последнего списка, включающего и компоненты в $y(1)$. Решение этой системы — заключительный шаг в линеаризационной атаке. Её сложность, как видим, равна 2^n, что много меньше 2^{n+m} — сложности атаки грубой силы. Кроме того, показанная возможность вычисления $y(1)$ по $x(1)$ означает, что при наличии в ключе вектора $x(1)$ добавление к ключу вектора $y(1)$ не повышает стойкости любого генератора G. Согласно [1, с. 331], для генераторов (δ, τ)-шагов этот факт давно известен. Здесь он установлен для более широкого класса криптографических генераторов.

4. Метод DSS

Система уравнений S называется *рекурсивно легко решаемой системой* (коротко РЛРС), если в ней существует непустая подсистема уравнений S' с небольшим эффективным подмножеством (ныне это не более 3–4 десятков) переменных, такая, что подсистема уравнений $S \setminus S'$ подстановкой в неё любого решения подсистемы S' преобразуется в РЛРС. По определению, такая система решается кратным применением метода DS к её подсистеме и подстановки полученных решений подсистемы в её дополнение. Для удобства дальнейших ссылок данный метод решения РЛРС назовём DSS (Devide, Solve and Substitute) — «разделяй, решай и подставляй».

Теперь можно заметить, что подсистема $E'(l)$, состоящая из трёх уравнений в $E(l)$ — четвёртого, пятого и шестого, т. е. подсистема уравнений управляемого автомата A_2, при любом фиксировании значений переменных в $y(1)$ становится РЛРС с переменными $u(t)$ для $t = 1, 2, \ldots, l$ и $y(t)$ для $t = 2, 3, \ldots, l$ и может быть решена методом DSS. В самом деле, при $t = 1$ и при фиксированном значении $y(1)$ имеем

уравнение

$$z(1) = f_2(u(1), y(1))$$

относительно $u(1)$, которое имеет либо два решения (0 и 1), если $z(1) = f_2(0, y(1)) = f_2(1, y(1))$, либо одно решение c, если $z(1) = f_2(c, y(1)) \neq f_2(\neg c, y(1))$, и не имеет решений в оставшемся случае $z(1) \neq f_2(0, y(1)) = f_2(1, y(1))$.

При $t = 2$, используя найденное на предыдущем шаге значение $u(1)$, вычисляем $y(2) = \neg u(1) g^{\delta}(y(1)) + u(1) g^{\tau}(y(1))$ и получаем уравнение

$$z(2) = f_2(u(2), y(2))$$

относительно $u(2)$, которое также может иметь два, одно или ни одного решения. Продолжая процесс для $t = 3, \ldots, l$, получим последовательности $u(1), \ldots, u(l)$ и $y(1), \ldots, y(l)$, которые являются кандидатами соответственно на выходную последовательность управляющего автомата и последовательность состояний управляемого автомата при известных $y(1), g_2, f_2$ и выходной последовательности $z(1), z(2), \ldots, z(l)$ генератора G.

Если начальное состояние $y(1)$ автомата A_2 не входит в ключ генератора, то криптоаналитику известно его значение и он может решить $E'(l)$ методом DSS, как описано выше, и найти возможные значения для $u(1), u(2), \ldots, u(l)$ на входе управляемого автомата A_2, которые являются возможными значениями функции f_1 на наборах $x(t)$ значений её переменных $x_1(t)x_2(t)\ldots x_n(t)$ для некоторых $t \in \{1, 2, \ldots, l\}$. Сложность этого вычисления полиномиальная, подробно процедура описана далее в п. 5 (алгоритм 1).

Если же состояние $y(1)$ автомата A_2 принадлежит ключу, то криптоаналитик может применить к системе $E'(l)$ метод DS, а именно: фиксируя в $E'(l)$ поочерёдно возможные значения состояния $y(1)$ и получая каждый раз некоторую РЛРС, найти такое значение $y(1)$, при котором полученная РЛРС совместна, решить её методом DSS и в результате получить и ключевое значение $y(1)$, и некоторые значения для $u(1), u(2), \ldots, u(l)$ функции f_1 на соответствующих наборах $x(t)$ значений её переменных. Сложность этой атаки равна 2^m. Её реализация сводится к последовательному применению алгоритма 1 для всех значений $y(1) \in \mathbb{F}_2^m$.

Таким образом, со сложностью не более 2^m криптоаналитик независимо от того, входит или не входит $y(1)$ в ключ, но зная f_2 и g_2 (а вместе со вторым и δ, τ, g), может получить доступ к некоторым значениям в отрезке $\beta = u(1)u(2)\ldots u(l)$ выходной последовательности автомата A_1 и тем самым открыть две возможности для решения задачи криптоанализа произвольного генератора G: 1) вычислить его ключ $(x(1), y(1))$ атакой «встреча посредине» и 2) свести её к криптоанализу управляющего автомата A_1, т. е. к определению ключа последнего по его выходноу отрезку β. Далее эти возможности рассматриваются в п. 6 и 7 соответственно.

5. Алгоритм DSS

Задача — методом DSS определить некоторые значения ключевой функции f_1 в случае, когда остальные параметры генератора известны. Алгоритм состоит в построении графа, вершины которого расположены по ярусам с номерами $t \in \{1, 2, \ldots, l\}$, вершины помечены натуральными числами, дуги — значениями 0 и 1.

Алгоритм 1. DSS

Вход: $z(1), \ldots, z(l)$; $x(1), y(1)$; $g, \sigma, \tau, g_1, f_2$
Выход: значения $f_1(x^{(i)})$, $i = 1, \ldots, k$, для некоторых $x^{(1)}, \ldots, x^{(k)} \in \mathbb{F}_2^n$

1: На ярусе 1 — одна вершина u (особая — её метка не имеет значения); $t := 2$.
2: **Если** $z(1) \neq f_2(0, y(1)) = f_2(1, y(1))$, **то**
3: выход из алгоритма с ответом «$y(1)$ не может быть начальным состоянием автомата A_2»,
4: **иначе**
5: на ярус 2 помещаем вершину v с меткой 1; соединяем вершину u с вершиной v двумя дугами (с метками 0 и 1), если $z(1) = f_2(0, y(1)) = f_2(1, y(1))$; одной дугой с меткой 0 (0-дугой), если $z(1) = f_2(0, y(1)) \neq f_2(1, y(1))$, и одной дугой с меткой 1 (1-дугой), если $z(1) = f_2(1, y(1)) \neq f_2(0, y(1))$.
6: **Если** вершин на ярусе t нет, **то**
7: выход из алгоритма с ответом «$y(1)$ не может быть начальным состоянием автомата A_2».
8: Рассматриваем каждую вершину v на ярусе t; пусть j — метка вершины v.
9: **Если** $z(t+1) = f_2(0, g^{\sigma+j}(y(1))) = f_2(1, g^{\tau+j}(y(1)))$, **то**
10: к вершине v добавляем два потомка на ярусе $t+1$: 0-дуга соединяет вершину v с вершиной, помеченной $\sigma + j$; 1-дуга — вершину v с вершиной, помеченной $\tau + j$.
11: **Если** $z(t+1) \neq f_2(0, g^{\sigma+j}(y(1))) = f_2(1, g^{\tau+j}(y(1)))$, **то**
12: потомков у вершины v нет; удаляем вершину v и дуги, ведущие в неё; поднимаемся по ярусам вверх, удаляя по пути все вершины, не имеющие потомков, и дуги, ведущие в них. Если поднялись до яруса 1, то выход с ответом «$y(1)$ не может быть начальным состоянием автомата A_2».
13: **Если** $z(t+1) = f_2(0, g^{\sigma+j}(y(1))) \neq f_2(1, g^{\tau+j}(y(1)))$, **то**
14: к вершине v добавляем одного потомка с меткой $\sigma + j$; соединяем v с потомком 0-дугой.
15: **Если** $z(t+1) = f_2(1, g^{\tau+j}(y(1))) \neq f_2(0, g^{\sigma+j}(y(1)))$, **то**
16: к вершине v добавляем одного потомка с меткой $\tau + j$; соединяем v с потомком 1-дугой.
17: Вершины яруса t, имеющие одинаковые метки, отождествляем.
18: **Если** $t < l - 1$, **то**
19: увеличиваем t на 1, переходим к шагу 6.
20: Анализируем построенный граф.
21: **Для** $t = 1, \ldots, l-1$
22: **Если** все дуги, соединяющие вершины t-го и $(t+1)$-го ярусов, имеют одинаковую метку $c \in \{0, 1\}$, **то**
полагаем $f_1(g_1^{t-1}(x(1))) = c$.

Если начальное состояние $y(1)$ автомата A_2 входит в ключ генератора, то, применяя алгоритм 1 для всех значений $y(1) \in \mathbb{F}_2^m$, можно ожидать, что количество «кандидатов» на роль начального состояния автомата A_2 будет много меньше, чем 2^m; так, в среднем в 1/4 случаев алгоритм закончит работу на шаге 3; ещё примерно 1/4 значений отсеется на шаге 7 при $t = 2$ и т. д. Получение более точных оценок и их зависимости от параметров генератора составляет предмет дальнейших исследований.

6. Атака «встреча посредине»

Пусть ключом генератора G является пара $(x(1), y(1)) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ и $l > n$.

Для каждого значения a переменной $x(1)$ в $\mathbb{F}_2^n$, пользуясь системой уравнений E_1 для $t = 1, 2, \ldots, l$, вычислим $\alpha = a_1 a_2 \ldots a_l$ как её решение для набора переменных $u(1), u(2), \ldots, u(l)$ и сохраним a по адресу $H(\alpha)$, где $H : \mathbb{F}_2^l \to \mathbb{F}_2^n$ есть некоторая хеш-функция. Эти действия выполняются предварительно, т. е. до проведения атаки, и их сложность не входит в её сложность. Во время атаки, имея выходную последовательность генератора $\gamma = z(1)z(2)\ldots z(l)$, методом DSS решается система E_2 уравнений автомата A_2 при разных значениях b переменной $y(1)$, выбираемых в $\mathbb{F}_2^m$ до тех пор, пока при некотором b в системе для набора переменных $u(1), u(2), \ldots, u(l)$ не будет получено решение $\beta = b_1 b_2 \ldots b_l$ с непустым содержимым a по адресу $H(\beta)$, и тогда пара (a, b) принимается за результат криптоанализа — искомое значение ключа. Сложность этой атаки не превосходит 2^m.

В альтернативном варианте атаки «встреча посредине» эти два её шага меняются местами, а именно: сначала по известной γ для каждого значения b переменной $y(1)$ методом DSS вычисляется β и значение b сохраняется по адресу $H(\beta)$ для некоторой хеш-функции $H : \mathbb{F}_2^l \to \mathbb{F}_2^m$, а затем последовательно для различных значений a переменной $x(1)$ вычисляются соответствующие значения α до тех пор, пока по адресу $H(\alpha)$ не будет найдено непустое b, и тогда пара (a, b) принимается за результат криптоанализа. Сложность этого варианта атаки не превосходит $2^m + 2^n$.

Разумеется, в каждом из вариантов атаки результат криптоанализа может быть ошибочным, т. е. не эквивалентным истинному ключу, но вероятность этого с ростом длины l отрезка γ может только падать.

7. Криптоанализ управляющего автомата

Итак, требуется по заданной выходной последовательности $u(1)u(2)\ldots u(l)$ автомата A_1 вычислить его ключ, которым в произвольном случае может быть любое подмножество в $\{x(1), g_1, f_1\}$. Если ключ есть $\{x(1)\}$, то $x(1)$ может быть вычислено как решение подсистемы E_1 уравнений A_1. Это может быть сделано любым подходящим методом [4], либо методом DS при наличии в E_1 малого эффективного множества переменных, в частности — линеаризационной атакой при наличии в E_1 малого линеаризационного множества переменных, либо методом грубой силы при малом n. Ясно, что сложность любой такой атаки не превосходит 2^n.

В случае, когда ключом генератора A_1 является его функция выходов f_1, принадлежащая некоторому классу C_1 булевых функций от n переменных, его криптоанализ, в свою очередь, сводится к доопределению в классе C_1 функции f', частично определённой соотношениями $u(t) = f'(x_1(t), x_2(t), \ldots, x_n(t))$, $t = 1, 2, \ldots, l$. Нетривиальные методы решения этой последней задачи в значительной степени определяются заданным классом C_1, который, опять же в свою очередь, может определяться ограничениями на сложностные характеристики функций в классе — на количество существенных аргументов, мощность эффективного или линеаризационного множества аргументов (любое фиксирование которых превращает функцию в вычислимую с полиномиальной сложностью или линейную соответственно), длину АНФ или ДНФ, степени мономов или ранги элементарных конъюнкций в них и т. п., или криптографическими свойствами этих функций [5, 6] — сбалансированностью, корреляционной иммунностью, устойчивостью, алгебраической иммунностью и др. Подобных классов булевых функций, в том числе полезных для практических применений в генераторах A_1 и интересных в теории криптоанализа таких генераторов, можно указать великое множество.

8. Атака на функцию выходов управляемого автомата

В случае, когда ключом генератора G является функция f_2, криптоаналитик, зная, а возможно, и выбирая $x(1), y(1)$ и наблюдая $z(t)$, может вычислить $x(t+1) = g_1(x(t))$, $u(t) = f_1(x(t))$ и $f_2(u(t), y(t)) = z(t)$ для $t = 1, 2, \ldots, l$. Последние равенства определяют f_2 частично, и ввиду принадлежности f_2 к некоторому известному классу C_2 булевых функций от $m + 1$ переменных задача криптоанализа G теперь опять же сводится к доопределению имеющейся частичной булевой функции f', определённой соотношениями $f'(u(t), y(t)) = z(t)$, $t = 1, 2, \ldots, l$, до функции в классе C_2.

9. Доопределение частичной функции в классе функций с малым числом существенных переменных

Авторам известен только один класс C, функции которого уже применялись в реальных конечно-автоматных генераторах в качестве одновременно функций выходов и ключей, а сами эти генераторы исследовались на стойкость к криптоаналитическим атакам. Это было 50 лет назад с участием первого автора. Речь идёт о классе булевых функций, зависящих существенно от малого числа (по тем временам — до 10) аргументов, выбираемых случайным образом из большого числа (по тем временам — за сотню) переменных, в нашем случае представляющих собой компоненты состояния $x_1, x_2, \ldots, x_n$ в автомате A_1 или $y_1, y_2, \ldots, y_m$ в автомате A_2.

Были найдены конструктивные необходимые и достаточные условия единственности доопределения части искомой функции в этом классе и разработан метод, позволяющий построить любое из существующих таких доопределений. Условия единственности и метод построения сформулированы в предположении, что криптоаналитику известно, кроме прочего, либо количество k существенных аргументов ключевой функции, либо его верхняя граница k_0. Метод построения решает свою задачу за два шага: сначала узнаёт существенные аргументы функции, затем — её значения на наборах значений этих аргументов. Для нахождения первых используются методы построения столбцовых покрытий булевой матрицы, имеющей строками покомпонентные разности наборов значений переменных искомой функции с разными значениями на них.

В полном изложении эти результаты опубликованы в [7]. На их основе сформулированы оптимальные алгоритмы криптоанализа, в которых последовательность $z(1)z(2)\ldots z(l)$ на выходе генератора наблюдается до такого наименьшего $l = 1, 2, \ldots$, при котором выполняются условия единственности доопределения, после чего вся функция строится данным методом. Результаты экспериментальных исследований этих алгоритмов на современных компьютерах, выражающие зависимости времени вычислений алгоритмами ключа и длины l потребного для этого отрезка выходной последовательности генератора от его параметров (n, k, k_0), представлены в работе [8].

ЛИТЕРАТУРА

1. *Фомичёв В. М.* Дискретная математика и криптология. М.: ДИАЛОГ-МИФИ, 2003. 400 с.
2. *Агибалов Г. П., Панкратова И. А.* К криптоанализу двухкаскадных конечно-автоматных криптографических генераторов // Прикладная дискретная математика. Приложение. 2016. № 9. С. 41–43.
3. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского государственного университета. Приложение. Сентябрь 2003. № 6. С. 31–41.

4. *Агибалов Г. П.* Методы решения систем полиномиальных уравнений над конечным полем // Вестник Томского государственного университета. Приложение. Август 2006. № 17. С. 4–9.
5. *Панкратова И. А.* Булевы функции в криптографии: учеб. пособие. Томск: Издательский Дом Томского государственного университета, 2014. 88 с.
6. *Токарева Н. Н.* Симметричная криптография. Краткий курс: учеб. пособие. Новосибирск: Изд-во Новосиб. ун-та, 2012. 234 с.
7. *Агибалов Г. П.* О некоторых доопределениях частичной булевой функции // Труды Сибирского физико-технического института. 1970. Вып. 49. С. 12–19.
8. *Агибалов Г. П., Сунгурова О. Г.* Криптоанализ конечно-автоматного генератора ключевого потока с функцией выходов в качестве ключа // Вестник Томского государственного университета. Приложение. Август 2006. № 17. С. 104–108.

REFERENCES

1. *Fomichev V. M.* Diskretnaya matematika i kriptologiya [Discrete Mathematics and Cryptology]. Moscow, DIALOG-MEPhI Publ., 2003. 400 p. (in Russian)
2. *Agibalov G. P. and Pankratova I. A.* K kriptoanalizu dvukhkaskadnykh konechno-avtomatnykh kriptograficheskikh generatorov [To cryptanalysis of 2-cascade finite automata cryptographic generators]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2016, no. 9, pp. 41–43.
3. *Agibalov G. P.* Logicheskie uravneniya v kriptoanalize generatorov klyuchevogo potoka [Logical equations in cryptanalysis of key stream generators]. Vestnik TSU. Prilozhenie, 2003, no. 6, pp. 31–41. (in Russian)
4. *Agibalov G. P.* Metody resheniya sistem polinomial'nykh uravneniy nad konechnym polem [Methods for solving systems of polynomial equations over a finite field]. Vestnik TSU. Prilozhenie, 2006, no. 17, pp. 4–9. (in Russian)
5. *Pankratova I. A.* Bulevy funktsii v kriptografii: ucheb. posobie [Boolean Functions in Cryptography: a Tutorial]. Tomsk, TSU Publ., 2014. 88 p. (in Russian)
6. *Tokareva N. N.* Simmetrichnaya kriptografiya. Kratkiy kurs: ucheb. posobie [Symmetric Cryptography. Short Course: a Tutorial]. Novosibirsk, NSU Publ., 2012. 234 p. (in Russian)
7. *Agibalov G. P.* O nekotorykh doopredeleniyakh chastichnoy bulevoy funktsii [Some completions of partial Boolean function]. Trudy SPhTI, 1970, iss. 49, pp. 12–19. (in Russian)
8. *Agibalov G. P. and Sungurova O. G.* Kriptoanaliz konechno-avtomatnogo generatora klyuchevogo potoka s funktsiey vykhodov v kachestve klyucha [Cryptanalysis of a finite-state keystream generator with an output function as a key]. Vestnik TSU. Prilozhenie, 2006, no. 17, pp. 104–108. (in Russian)

УДК 519.719.2

ОБЗОР АТАК НА AES-128: К ПЯТНАДЦАТИЛЕТИЮ СТАНДАРТА AES

К. Д. Жуков

Лаборатория ТВП, г. Москва, Россия

Представлен обзор работ, опубликованных до 2016 г. и посвящённых криптоанализу алгоритма AES-128 (Advanced Encryption Standard). Перечислены основные криптографические методы, используемые при анализе AES. Приведены сложностные характеристики 88 атак на редуцированные варианты алгоритма AES-128. Указано необходимое для проведения атак количество известных пар шифрованных и открытых текстов с условиями на них. В поле зрения не попали атаки по побочным каналам и атаки с ограничением на используемые ключи.

Ключевые слова: *AES, Advanced Encryption Standard, методы дешифрования.*

DOI 10.17223/20710410/35/5

OVERVIEW OF ATTACKS ON AES-128: TO THE 15^{TH} ANNIVERSARY OF AES

K. D. Zhukov

TVP Laboratory, Moscow, Russia

E-mail: zhukov_kirill@bk.ru

This overview covers attacks on the reduced AES-128 published up to the end of 2016. We enumerate main cryptographic methods used in cryptanalysis of AES. We also tabulate the complexity characteristics of 88 attacks on the reduced AES-128 including the number and peculiarities of necessary plaintexts and ciphertexts. Side-channel attacks and related key attacks are out of the overview scope.

Keywords: *AES, Advanced Encryption Standard, key-recovery attack.*

Введение

В 1998 г. была опубликована схема блочного шифра Rijndael. С тех пор начался его интенсивный криптоанализ, который ещё больше усилился с принятием алгоритма в качестве американского стандарта шифрования AES [1]. За 15 лет опубликовано более ста печатных работ, посвящённых анализу Rijndael. Первая атака, теоретически понижающая стойкость AES, опубликована лишь в 2011 г. [2], спустя 10 лет после публикации стандарта. При этом понижение стойкости составило не более чем три двоичных порядка.

В общем случае под атаками на систему шифрования понимаются не только методы дешифрования. Например, в литературе изучается задача идентификации AES [3, 4]. В некоторых случаях на основе методов идентификации формулируются методы дешифрования. Далее сосредоточимся на работах, содержащих методы дешифрования.

Можно выделить три основных предположения криптоанализа, в рамках которых проводятся атаки на AES:

— анализ видоизменённой схемы (например, снижение числа раундов);
— использование дополнительной информации (например, утечка по побочным каналам, внесение искажений);
— специфические условия эксплуатации (например, подобранные открытые или шифрованные тексты, связанные ключи).

Данный обзор посвящен атакам на AES с уменьшенным числом раундов. При анализе редуцированных алгоритмов наблюдаются два подхода. Первый заключается в теоретическом понижении стойкости (по сравнению с методом тотального опробования) редуцированной схемы. Второй подход нацелен на построение практических атак на редуцированные версии алгоритма. Два подхода в определённой степени независимы: атаки, понижающие стойкость большего числа раундов, могут не улучшать атак на малое число раундов.

1. Алгоритм AES и способы построения редуцированных схем

Описанию алгоритма AES и используемому в нём математическому аппарату посвящена работа [5]. Ниже приводится краткая схема шифра.

Алгоритм AES представляет собой блочный шифр с длиной блока 128 битов (16 байтов) и с тремя возможными длинами ключей — 128, 192 и 256 битов. Блок шифра (промежуточное состояние) представляют в виде квадратной матрицы над полем $\mathrm{GF}(2^8)$ размера 4. Каждый элемент записывается одним байтом. Алгоритм реализуется итеративно несколькими раундами, которые состоят из следующих преобразований:

— `SubBytes (SB)` — применение одного и того же обратимого преобразования (S-блока) ко всем 16 байтам промежуточного состояния шифра;
— `ShiftRows (SR)` — циклический сдвиг каждой строки промежуточного состояния шифра (i-я строка сдвигается влево на i байтов, $i = 1, 2, 3, 4$);
— `MixColumns (MC)` — умножение слева каждого столбца промежуточного состояния на постоянную квадратную матрицу над $\mathrm{GF}(2^8)$ порядка 4;
— `AddRoundKey (ARK)` — побитовое сложение с раундовым ключом.

На первом раунде производится операция `AddRoundKey` с использованием дополнительного «отбеливающего» (*whitening*) ключа, а на последнем раунде отбрасывается операция `MixColumns`. Количество раундов зависит от длины используемого ключа: 10 раундов для 128-битового ключа, 12 раундов — для 192-битового и 14 раундов — для 256-битового ключа. Три варианта алгоритма AES обозначаются AES-128, AES-192 и AES-256 соответственно.

В работе [6] отмечается, что при одном и том же количестве раунодв AES стойкость итеративной схемы может существенно зависеть от наличия операции `MixColumns` на последнем раунде. Большинство известных атак на редуцированные версии AES сделаны в предположении, что на последнем раунде операция `MixColumns` не производится.

Однозначного определения редуцированного шифра AES нет даже у создателей стандарта. В исходной работе [7] редуцированные варианты AES, так же как и полный AES, не содержат операции `MixColumns` на последнем раунде шифрования. Одновременно с этим, при анализе редуцированного шифра в работе [8] операция `MixColumns` включается в последний раунд.

В [9] с помощью аппарата базисов Гребнера анализируются редуцированные схемы, в которых уменьшается количество строк (столбцов) в прямоугольном представлении промежуточного блока шифра, а также байты заменяются полубайтами. Такие редукции далее не рассматриваются.

Под атаками на AES будем понимать, в том числе, атаки на редуцированные варианты AES. Говоря об атаках на k раундов AES, будем предполагать, что на последнем раунде операция `MixColumns` не применяется. В противном случае будем говорить об атаках на k полных раундов AES.

2. Модели проведения атак на AES

Под моделью проведения атаки понимаются условия, в которых возможно реализовать атаку, а точнее, какие данные о работе шифра доступны криптоаналитику. В худшем случае наблюдателю доступны только шифрованные тексты на некоторых случайных ключах. В такой строгой модели формулирование атак затруднительно. Стойкость AES исследуется, как правило, в моделях из трёх основных классов: общие модели, модели с утечками по побочным каналам и модели с зависимыми ключами.

2.1. Общие модели

В общих моделях предполагается, что криптоаналитику известно некоторое количество пар открытых и соответствующих им шифрованных текстов. В классической постановке на открытые и шифрованные тексты не накладывается никаких ограничений. Наибольшее число атак на AES сделано в предположении, что открытые тексты, для которых известен результат шифрования, подобраны некоторым специальным образом. Реже встречаются атаки, в которых делается предположение о наличии открытых текстов для подобранных специальным образом шифрованных текстов.

В криптоанализе блочных шифров исследуются модели с адаптивно подбираемыми открытыми или шифрованными текстами [10]. Однако автор таких атак на AES не знает.

В работе [11] рассматриваются модели открытых текстов с подобранными разностями (при этом сами открытые тексты неизвестны).

2.2. Модели с утечками по побочным каналам

В последнее время становится популярным криптоанализ с использованием побочных каналов информации (*side-channel attacks*). Такую тенденцию можно объяснить двумя причинами. С одной стороны, с развитием технической базы атаки с использованием побочных каналов кажутся всё более и более реалистичными с точки зрения их реализации на практике. С другой стороны, при синтезе шифров такие атаки редко учитываются, а следовательно, у криптоаналитиков появляется возможность существенно понизить стойкость алгоритмов при некоторых благоприятных условиях.

Криптоанализ AES с использованием побочных каналов утечки информации представлен такими направлениями, как атаки при наличии доступа в кэш-память процессора (*cache attacks* [12]), анализ энергопотребления (*power analysis* [13]) и анализ разностных искажений (*differential fault analysis* [14]). Последний предполагает наличие возможности активного вмешательства в процедуру шифрования.

В данном обзоре атаки по побочным каналам на AES не рассматриваются. Однако нужно отметить, что методы разностных искажений могут быть применимы для построения атак в общих моделях. В [11] показано, что методы разностных искажений и разностный криптоанализ тесно связаны. Внесение искажения перед первым раундом шифрования можно рассматривать как подобранную пару текстов с фиксированной разностью в общей модели.

2.3. Модели с зависимыми ключами

В моделях с зависимыми ключами (*related key attack*) считается, что криптоаналитику известно некоторое количество пар открытых и шифрованных текстов, полученных путём шифрования на ключах, связанных каким-либо соотношением.

Как правило, в данной модели строится атака на такую пару ключей K_1 и K_2, что $K_2 = K_1 \oplus C$, где C — заданная константа. Применительно к AES существуют атаки и для более сложных зависимостей. Например, в [15] исследуется случай $K_2 = F^{-1}(F(K_1) \oplus C)$, где отображение F представляет собой один раунд ключевого расписания AES-256 (*related subkey attack*). В данном обзоре атаки с зависимыми ключами не рассматриваются.

3. Некоторые математическим методы криптоанализа AES

Ниже перечислены несколько математических методов, используемых при анализе AES в общей модели. Для некоторых методов приводится краткая идея или основное свойства шифра, позволяющее применить метод. Отметим, что приводимый список не претендует на полноту и затрагивает наиболее популярные техники.

3.1. Разностный криптоанализ

В общем случае разностный метод не эффективен для практических атак на AES, так как уже для четырёх раундов максимальное разностное отклонение не больше 2^{-150}. Однако данный метод и его модификации применяются для теоретического понижения стойкости редуцированных версий алгоритма.

Метод запретных разностей

Метод запретных разностей (*impossible differentials*) основан на следующем свойстве алгоритма AES [16].

Свойство 1. Два открытых текста, отличающихся только в одном байте, не могут при шифровании четырьмя раундами AES перейти в шифртексты с совпадающими байтами под номерами 1, 6, 11 и 16.

Зачастую метод запретных разностей применяется вместе с методом «встреча посередине».

Метод усеченных разностей

Метод усечённых разностей (*truncated differentials*) предложен в [17] в 1995 г. В отличие от классического разностного криптоанализа, в методе исследуются разности не блоков, а только частей блоков. Атаки на AES с применением данного метода построены в [4].

Техника раннего обрыва

В [18] метод запретных разностей усовершенствован с использованием техники раннего обрыва (*early abort technique*), которая опирается на следующее свойство AES.

Свойство 2. Рассмотрим набор из всевозможных 4×255 промежуточных состояний шифра с ненулевым байтом в первой колонке. После последовательного применения преобразований $\mathtt{MC}^{-1}$ и $\mathtt{RC}^{-1}$ пара байтов под номерами 1 и 6 будет различна во всех 4×255 состояниях набора.

3.2. Политопный криптоанализ

Политопный криптоанализ появился в 2016 г. как обобщение разностного метода. В [19] с помощью метода запретных политопных переходов получены атаки на редуцированный AES.

3.3. Интегральный криптоанализ

Авторство интегрального криптоанализа принадлежит Л. Кнудсену. С помощью данного подхода изначально анализировался предшественник алгоритма Rijndael — блочный шифр Square [20]. Отсюда берёт название базовая атака интегрального криптоанализа применительно к анализу AES — атака «Квадрат».

Метод «Квадрат»

Метод основан на следующем свойстве блочного шифра AES [7].

Свойство 3. Рассмотрим набор из 256 открытых текстов, в которых на фиксированной позиции байты пробегают всевозможные 256 значений, а на остальных позициях во всех текстах байты одинаковы. После применения трёх раундов AES побитовая сумма всех полученных 256 шифртекстов равна 0.

Метод частичных сумм

В 2000 г. в [21] предложена модификация атаки «Квадрат», получившая название техники частичных сумм (*partial sums*), которая была затем улучшена в работе [22].

3.4. Метод цепочек подпространств

Метод цепочек подпространств (*subspace trail*) появился в 2016 г. [4] как обобщение известного метода инвариантных подпространств [23]. Для построения атак на AES метод комбинируется с вариациями разностного метода.

3.5. Метод «Биклик»

Метод «Биклик» впервые применён к анализу AES в 2011 г. [2] А. Богдановым, Д. Хорватовичем и К. Рехбергом. В настоящее время метод является самым сильным при анализе полнораундовых вариантов алгоритма AES.

3.6. Метод «встреча посередине»

Метод «встреча посередине» предлагает общий подход к анализу блочных шифров. Данная техника может применяться в совокупности с другими методами. В общем виде с методом можно ознакомиться, например, в [2, 24].

Использование метода «встреча посередине» на отдельных подэтапах восстановления ключа

В [25] показано, что применение метода «встреча посередине» позволяет сформулировать эффективные критерии отбраковки ложных вариантов ключа на определённых этапах его восстановления в редуцированном алгоритме AES.

Техника просеивания посередине

В 2013 г. французские математики предложили технику просеивания посередине (*sieve-in-the-middle*) [24]. Техника получила развитие в работе [26], где применяется вместе с методом «Биклик». Отметим, что похожая техника зависящего от ключа просеивания (*key-dependent sieve*) в менее общем виде сформулирована независимо китайскими учеными [27].

Метод построения коллизий

В 1999 г. в [28] впервые было предложено использовать методы идентификации блочного шифра Rijndael для построения атак на редуцированные версии алгоритма. Эти идеи получили развитие в работах азиатских ученых [29 – 32].

3.7. Автоматизированный поиск алгебраических соотношений и разностных характеристик

Французские криптографы разрабатывают автоматизированные (реализуемые с помощью компьютера) методы поиска алгебраических соотношений, связывающих биты открытого и шифрованного текстов с битами ключа [33], соотношений между битами раундовых ключей [34], а также поиска разностных характеристик для связанных ключей [35]

4. Обзор атак на редуцированный AES-128 в общей модели

На момент составления данного обзора опубликована не одна сотня атак на AES-128, AES-192 и AES-256. В таблице приводятся атаки только на редуцированный AES-128 в классической модели (атаки по побочным каналам и в модели с зависимыми ключами не рассматриваются). Общее количество атак на редуцированный AES-128 с учётом различных моделей приблизительно вдвое больше, чем указано в таблице.

В первом столбце таблицы для дальнейших ссылок указан порядковый номер атаки. Атаки упорядочены по количеству анализируемых раундов AES-128 (второй столбец). Атаки на одинаковое количество раундов перечислены в произвольном (не хронологическом) порядке.

Третий столбец содержит индикатор присутствия операции `MixColumns` на последнем раунде рассматриваемого варианта редуцированного шифра. Присутствие операции обозначается символом ✓, а отсутствие — символом ✗. Как отмечалось выше (п. 1), стойкость редуцированной схемы существенно зависит от наличия операции `MixColumns`.

В четвёртом столбце указан объём материала, необходимый для проведения атаки. Под объёмом материала понимается количество пар открытых и соответствующих им шифрованных текстов, известных атакующему. Атаки на AES-128 используют один из трёх возможных типов необходимого материала: произвольные известные открытые тексты (ИО), подобранные открытые тексты (ПО) или подобранные шифрованные тексты (ПШ). Тип материала указан в пятом столбце.

В шестом столбце указывается емкостная сложность атак, измеряемая в блоках AES-128 (длина одного блока составляет 16 байтов). Отметим, что некоторые авторы атак указывают объём используемой памяти приближённо. Например, в [33] приводятся атаки, требующие памяти порядка одного блока AES-128. В данном обзоре такие оценки не уточняются и приводятся в соответствии с источником.

В седьмом столбце указана вычислительная сложность (или трудоёмкость) атак. Следуя общепринятой терминологии, трудоёмкость атаки на k раундов AES-128 ($k = 1, 2, \ldots, 10$) будем измерять в операциях зашифрования (или эквивалентно — расшифрования) одного 16-байтового блока с помощью k раундов AES-128. В редких случаях обращение в память (*memory access*) является основной элементарной операцией атаки. Для таких атак трудоёмкость измеряется в количестве операций доступа в память (ДП). Отметим, что некоторые атаки требуют проведения предварительных вычислений. Вычисления, которые необходимо произвести один раз, позволяют многократно атаковать шифр с меньшей трудоёмкостью. Например, трудоёмкость предварительного этапа атаки 68 составляет 2^{123}, а оперативного — 2^{113}. В сравнительной таблице трудоёмкость указана в худшем случае, в том числе с учётом предварительного этапа.

В восьмом столбце указан основной математический метод, используемый в атаке (или тип атаки). Типы атак, указанные в таблицах, являются условными. Как отме-

чалось выше, многие атаки совмещают несколько различных подходов. Кроме того, даже предлагаемые авторами классификации своих атак не всегда соответствуют распространённому пониманию тех или иных методов. Например, атаки 1, 6 и 17 отнесены их авторами к методу «встреча посередине», хотя они и не используют память. Тем не менее в таблицах указывается тип атаки в соответствии с источником.

В девятом столбце таблицы указана ссылка на работу, в которой предложена атака.

В некоторых работах указаны не все характеристики. Например, многие авторы зачастую не приводят оценок емкостных сложностей своих атак. Получение таких характеристик не всегда тривиально и может быть связано с детальным изучением отдельно взятых атак. В этих случаях в таблице стоит прочерк. Цветом выделены атаки, которые реализованы их авторами программно. В некоторых случаях исходные коды программ доступны в сети Интернет.

Атаки на редуцированный AES-128

№	Раунд	MC	Объём мат.	Тип мат.	Память	Трудоём.	Тип атаки	Работа
1	2	3	4	5	6	7	8	9
1	1	✓	1	ИО	1	2^{40}	Встр. посеред.	[25]
2	1	✓	1	ИО	2^{24}	2^{32}	Встр. посеред.	[25]
3	1	✓	2	ИО	1	2^{12}	Разностн.	[25]
4	1	✓	1	ИО	—	2^{48}	—	[6]
5	1	✓	1	ИО	2^{32}	2^{16}	—	[33]
6	2	✓	1	ИО	1	2^{80}	Встр. посеред.	[25]
7	2	✓	1	ИО	2^{49}	2^{64}	Встр. посеред.	[25]
8	2	✓	2	ИО	1	2^{48}	Разностн.	[25]
9	2	✓	2	ПО	1	2^{28}	Разностн.	[25]
10	2	✓	3	ИО	1	2^{32}	Разностн.	[25]
11	2	✗	2	ПО	—	2^{29}	Разностн.	[36]
12	2	✗	1	ИО	1	2^{56}	—	[33]
13	2	✗	2	ИО	2^{16}	2^{24}	—	[33]
14	2	✓	1	ИО	2^{48}	2^{64}	—	[33]
15	2	✓	2	ИО	2^{24}	2^{32}	Встр. посеред.	[33]
16	2	✓	2	ПО	2^{8}	2^{8}	—	[33]
17	3	✓	1	ИО	1	2^{120}	Встр. посеред.	[25]
18	3	✓	1	ИО	2^{49}	2^{104}	Встр. посеред.	[25]
19	3	✓	2	ПО	1	2^{32}	Разностн.	[25]
20	3	✓	9	ИО	2^{35}	2^{40}	Разн., Встр.	[25]
21	3	✗	1	ИО	2^{88}	2^{88}	—	[33]
22	3	✗	2	ИО	2^{80}	2^{80}	—	[33]
23	3	✗	2	ПО	2^{16}	2^{24}	—	[33]
24	3	✓	1	ИО	2^{72}	2^{96}	—	[33]
25	3	✓	2	ПО	2^{8}	2^{16}	—	[33]
26	3	✗, ✓	2	ПО	2^{8}	$2^{31,55}+2^{32}$ДП	Усеч. разност.	[4]
27	3	✗, ✓	3	ПО	1	$2^{11,2}$	Усеч. разност.	[4]
28	3	✗, ✓	3	ПО	2^{12}	$2^{5,1}+2^{10}$ДП	Усеч. разност.	[4]
29	4	✗	2^{9}	ПО	1	2^{9}	Квадрат	[7]
30	4	✓	2	ПО	1	2^{104}	Разн., Встр.	[25]
31	4	✓	5	ПО	2^{68}	2^{64}	Разн., Встр.	[25]
32	4	✓	10	ПО	2^{43}	2^{40}	Разн., Встр.	[25]
33	4	✗	12	ПО	—	2^{55}	Разностн.	[36]
34	4	✗	30	ПО	—	2^{54}	Разностн.	[36]
35	4	✗	2^{11}	ПО	—	2^{52}	Разностн.	[36]
36	4	✓	1	ИО	2^{80}	2^{120}	Встр. посеред.	[33]
37	4	✓	2	ПО	2^{80}	2^{80}	Встр. посеред.	[33]
38	4	✓	2^{2}	ПО	2^{24}	2^{32}	Встр. посеред.	[33]
39	4	✗	2^{9}	ПО	—	2^{8}	Квадрат	[16]

Окончание таблицы

№	Раунд	МС	Объём мат.	Тип мат.	Память	Трудоём.	Тип атаки	Работа
1	2	3	4	5	6	7	8	9
40	4	✗, ✓	2^3	ПО	2^{15}	2^{38}	Запрет. полит.	[19]
41	4	✗, ✓	2	ПО	1	2^{96}	Усеч. разност.	[4]
42	4	✗, ✓	3	ПО	1	$2^{74,7}$	Усеч. разност.	[4]
43	4	✗, ✓	3	ПО	2^{12}	$2^{64}+2^{76}$ДП	Усеч. разност.	[4]
44	4	✗, ✓	24	ПО	2^{17}	$2^{33,9}+2^{40,6}$ДП	Усеч. разност.	[4]
45	5	✗	2^{11}	ПО	1	2^{40}	Квадрат	[7]
46	5	✗	2^{32}	ПО	2^{32}	2^{40}	Квадрат	[7]
47	5	✗	2^{11}	ПО	—	2^{39}	Квадрат	[16]
48	5	✗	$2^{29,5}$	ПО	—	2^{31}	Запрет. разн.	[16]
49	5	✗	2^8	ПО	—	$2^{37,5}$	Квадрат	[36]
50	5	✗	2^8	ПО	—	2^{38}	Частич. сумм.	[22]
51	5	✗, ✓	15	ПО	2^{41}	2^{70}	Запрет. полит.	[19]
52	6	✗	$6 \cdot 2^{32}$	ПО	—	2^{44}	Частич. сумм.	[21]
53	6	✗	$2^{108,5}$	ИО	—	2^{120}	Разн., Встр.	[25]
54	6	✗	2^{32}	ПО	2^{32}	2^{72}	Квадрат	[7]
55	6	✗	2^{32}	ПО	—	2^{71}	Квадрат	[16]
56	6	✗	$2^{91,5}$	ПО	2^{89}	2^{122}	Запрет. разн.	[37]
57	6	—	$2^{114,5}$	ПО	—	2^{50}	Запрет. разн.	[38]
58	6	—	$2^{75,5}$	ПО	—	2^{104}	Запрет. разн.	[38]
59	6	✗	2^{32}	ПО	2^{40}	2^{42}	Частич. сумм.	[22]
60	6	✗	$2 \cdot 2^{32}$	ПО	—	—	Частич. сумм.	[39]
61	6	✗	2^8	ПО	$2^{106,17}$	$2^{106,17}$	Встр. посеред.	[34]
62	7	✗	2^{32}	ПО	2^{32}	$\approx 2^{128}$	Встр. (коллиз.)	[28]
63	7	✗	$2^{128}-2^{119}$	ПО	2^{64}	2^{120}	Частич. сумм.	[21]
64	7	✗	$2^{117,5}$	ПО	2^{105}	2^{123}	Запрет. разн.	[40]
65	7	✗	$2^{112,2}$	ПО	—	$2^{117,2}$ ДП	Запрет. разн.	[18]
66	7	—	$2^{115,5}$	ПО	2^{41}	2^{119}	Запрет. разн.	[38]
67	7	—	$2^{115,5}$	ПО	2^{105}	2^{119}	Запрет. разн.	[41]
68	7	—	2^{80}	ПО	2^{122}	2^{123}	Встр. посеред.	[30]
69	7	—	$2^{115,32}$	ПО	—	$2^{119,32}$	Запр. разн.	[42]
70	7	✗	2^{103+k}	ПО	2^{103+k}	2^{129-k}	Встр. (коллиз.)	[43]
71	7	—	$2^{106,2}$	ПО	$2^{90,2}$	$2^{107,1}+2^{117,2}$ДП	Запрет. разн.	[44]
72	7	✗	2^{105}	ПО	2^{90}	2^{99}	Встр. посеред.	[45]
73	7	✗	2^{99}	ПО	2^{96}	2^{99}	Встр. посеред.	[45]
74	7	—	2^{80}	ПО	2^{65}	2^{127}	Запрет. разн.	[46]
75	7	✗	2^{32}	ПО	$2^{126,47}$	$2^{126,47}$	Встр. посеред.	[34]
76	8	✗	$2^{126,33}$	ПШ	2^{102}	$2^{124,97}$	Биклик	[2]
77	8	✗	2^{127}	ПШ	2^{32}	$2^{125,64}$	Биклик	[2]
78	8	✗	2^{88}	ПШ	2^8	$2^{125,34}$	Биклик	[2]
79	10	✗	2^4	ПО	2^8	$2^{126,89}$	Биклик	[47]
80	10	✗	2^{88}	ПШ	2^8	$2^{126,18}$	Биклик	[2]
81	10	✗	2^{88}	ПШ	2^{64}	$2^{125,69}$	Биклик, Прос.	[24]
82	10	✗	2^{128}	ИО	2^8	$2^{125,56}$	Биклик	[26]
83	10	✗	2^{128}	ИО	2^{64}	$2^{125,35}$	Биклик, Прос.	[26]
84	10	✗	2^{64}	ПШ	2^8	$2^{126,16}$	Биклик	[26]
85	10	✗	2^{64}	ПШ	2^{64}	$2^{125,98}$	Биклик, Прос.	[26]
86	10	✗	2	ИО	2^8	$2^{126,67}$	Биклик	[26]
87	10	✗	2	ИО	2^{64}	$2^{126,59}$	Биклик, Прос.	[26]
88	10	✗	2^{24}	ПШ	2^9	$2^{126,5}$	Биклик	[48]

Численные характеристики атак приведены в соответствии с первоисточником, за некоторым исключением. Из работы [44] взяты оценки емкостной сложности атак 56, 63 и 66, а также все характеристики атаки 67. Из работы [26] взяты характеристики атак 81 и 79. В соответствии с [4] приводится информация о том, что трудоёмкость

атак 40 и 51 не зависит от наличия операции `MixColumns` на последнем раунде редуцированного шифра.

Особо подчеркнём важность обращения к первоисточникам при цитировании характеристик атак. Некоторые авторы, ссылаясь на более ранние атаки, указывают характеристики, отличные от приведённых в исходных работах. Примеры публикаций новых оценок характеристик более ранних атак можно найти в работах М. Танстолла [22, 36] и китайских учёных [38]. Как правило, комментариев к подобным оценкам не имеется. Можно предположить, что новые оценки получены в результате более тщательного анализа трудоёмкости.

Необходимо также отметить, что численные оценки характеристик атак являются субъективными и иногда имеют нестрогое обоснование. Зашифрование (расшифрование) одного блока AES-128 зачастую не является основной элементарной операцией атаки и пересчёт производится с определёнными допущениями. Например, в [28] приводится атака (см. атаку 62 в таблице) с опробованием всех 2^{128} вариантов ключа. По заявлению авторов, операция отбраковки одного варианта ключа требует меньше времени, чем одна операция зашифрования (расшифрования) AES-128. В соответствии с [38] данная трудоёмкость оценивается как $\approx 2^{128}$.

Другим примером неоднозначности характеристик атак на AES служит оценка вычислительной трудоёмкости атаки 81. В исходной работе [24] не приводится точной оценки трудоёмкости атаки. Однако в [26] авторы метода «Биклик» цитируют атаку 81 с оценкой трудоёмкости в $2^{125{,}69}$ операций шифрования, при этом отмечая, что по их подсчётам трудоёмкость составляет $2^{125{,}98}$.

5. Сравнение атак на AES-128 и атаки с практической трудоёмкостью

Тенденция развития методов анализа AES такова, что большинство авторов стремится атаковать большее число раундов с трудоёмкостью, меньшей тотального опробования. Реже криптографы сосредотачиваются на атаках практической трудоёмкости на малое число раундов [11]. В данном обзоре не рассматривается вопрос о возможном применении методов понижения стойкости большого числа раундов AES-128 для формулирования атаки на меньшее число раундов с практической трудоёмкостью.

Как предложено в [11], будем говорить о практической атаке на редуцированный AES, если трудоёмкость не превышает 2^{56} операций шифрования.

В модели с одной известной парой открытого и шифрованного текстов с практической трудоёмкостью удаётся атаковать два раунда AES-128 (атака 12), а с трудоёмкостью, близкой к практической, — два полных раунда AES-128 (атака 14). Использование двух пар открытого и шифрованного текстов позволяет атаковать два полных раунда уже с практической трудоёмкостью (атака 15).

Если рассматривать модель с произвольным числом пар известных открытых и шифрованных текстов, то можно атаковать с практической трудоёмкостью максимально три полных раунда AES-128 (атака 20). Для этого потребуется всего девять пар шифрованных и открытых текстов. Атака комбинирует разностный метод и метод «встреча посередине».

В модели подобранных открытых текстов максимальное число раундов AES-128, атакуемых с практической трудоёмкостью, равно шести. Атака опубликована в 2014 г. (см. атаку 60). Авторы верифицировали атаку с помощью программной реализации предлагаемого метода. С использованием шести персональных компьютеров (с четырёхъядерным процессором и 8-гигабайтной оперативной памятью каждый) секретный

ключ был найден за 12 дней. Исходные коды программы, реализующей атаку, имеются в свободном доступе в сети Интернет.

На семь раундов AES-128 неизвестно атак с трудоёмкостью меньшей чем 2^{99} операций шифрования.

Большинство атак на AES-128 основывается на нескольких идеях и свойствах шифра, описанных в том числе ещё в исходной работе 1998 г., представляющей алгоритм Rijndael. Наиболее сильные атаки практической трудоёмкости используют метод «Квадрат». Наиболее сильные атаки теоретического понижения стойкости большого числа раундов AES основаны на методах «встреча посередине» и «Биклик» и их комбинациях. С использованием метода запретных разностей, как правило, формулируются наименее сильные атаки.

Заключение

Список литературы, посвящённой анализу американского стандарта шифрования, насчитывает несколько десятков работ, в каждой из которых описывается не один вариант атаки на AES при различных дополнительных условиях. Цель данной работы — составление наиболее полного обзора опубликованных атак.

Приведены характеристики 88 атак на редуцированные варианты AES-128. В поле зрения не попали атаки по побочным каналам и атаки с ограничением на используемые ключи. Как правило, атаки на AES-128 допускают обобщения для AES-192 и AES-256. В рамках данного обзора не проводилось подробное табулирование атак на два последних варианты алгоритма.

Самыми сильными атаками, понижающими теоретическую стойкость, являются атаки типа «Биклик», применённые впервые к AES в 2011 г. С помощью данного метода удаётся построить атаку на десять раундов AES-128 (нередуцированный вариант) с трудоёмкостью $2^{126{,}67}$ операций шифрования и с незначительным использованием памяти в модели с двумя известными парами открытого и шифрованного текстов.

Максимальное количество раундов AES-128, которое удаётся атаковать с практической трудоёмкостью, равно шести (атака опубликована в 2014 г.). Такое количество раундов достигается с помощью метода частичных сумм, поэтому для атаки необходимо 2^{33} подобранных открытых текстов.

В модели с несколькими известными парами открытого и шифрованного текстов с практической трудоёмкостью можно атаковать три полных раунда AES-128. Атака основана на комбинации разностного метода и метода «встреча посередине» и требует девяти пар открытого и шифрованного текстов.

ЛИТЕРАТУРА

1. `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf` — National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard. 2001.
2. *Bogdanov A., Khovratovich D., and Rechberger C.* Biclique cryptanalysis of the full AES // ASIACRYPT 2011. LNCS. 2011. V. 7073. P. 344–371.
3. *Gilbert H. and Peyrin T.* Super-sbox Cryptanalysis: Improved Attacks for AES-like Permutations. Cryptology ePrint Archive, Report 2009/531. 2009.
4. *Grassi L., Rechberger C., and Ronjom S.* Subspace Trail Cryptanalysis and its Applications to AES. Cryptology ePrint Archive, Report 2016/592. 2016.
5. *Daemen J. and Rijmen V.* The Design of Rijndael: AES — The Advanced Encryption Standard. Berlin: Springer, 2002. 238 p.

6. *Dunkelman O. and Keller N.* The effects of the omission of last round's mixcolumns on AES // Inform. Proc. Let. 2010. V. 110. No. 8–9. P. 304–308.
7. *Daemen J. and Rijmen V.* AES Proposal: Rijndael. 1998. `http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf`.
8. *Bouillaguet C., Derbez P., Dunkelman O., et al.* Low-data complexity attacks on AES // IEEE Trans. Inform. Theory. 2012. V. 58. No. 11. P. 7002–7017.
9. *Bulygin S. and Brickenstein M.* Obtaining and solving systems of equations in key variables only for the small variants of AES. Cryptology ePrint Archive, Report 2008/435. 2008.
10. *Van Tilborg H.* Encyclopedia of Cryptography and Security. Berlin: Springer, 2005. 684 p.
11. *Tunstall M.* Practical complexity differential cryptanalysis and fault analysis of AES // J. Cryptographic Eng. 2011. V. 1. No. 3. P. 219–230.
12. *Bogdanov A. and Pyshkin A.* Algebraic Side-Channel Collision Attacks on AES. Cryptology ePrint Archive, Report 2007/477. 2007.
13. *Osvik D. A., Shamir A., and Tromer E.* Cache Attacks and Countermeasures: the Case of AES. Cryptology ePrint Archive, Report 2005/271. 2005.
14. *Ali S. S., Mukhopadhyay D., and Tunstall M.* Differential Fault Analysis of AES: Towards Reaching its Limits. Cryptology ePrint Archive, Report 2012/446. 2012.
15. *Biryukov A., Dunkelman O., Keller N., et al.* Key Recovery Attacks of Practical Complexity on AES Variants with up to 10 Rounds. Cryptology ePrint Archive, Report 2009/374. 2009.
16. *Biham E. and Keller N.* Cryptanalysis of reduced variants of Rijndael // Proc. 3rd AES Conf. N. Y., 1999. P. 11–15.
17. *Knudsen L. R.* Truncated and higher order differentials // LNCS. 1995. V. 1008. P. 196–211.
18. *Lu J., Dunkelman O., Keller N., and Kim J.* New Impossible Differential Attacks on AES. Cryptology ePrint Archive, Report 2008/540. 2008.
19. *Tiessen T.* Polytopic cryptanalysis // Proc. 35th Ann. Intern. Conf. Advances in Cryptology — EUROCRYPT 2016. V. 9665. N. Y.: Springer, 2016. P. 214–239.
20. *Daemen J., Knudsen L., and Rijmen V.* The block cipher square // LNCS. 1997. V. 1267. P. 149–165.
21. *Ferguson N., Kelsey J., Lucks S., et al.* Improved cryptanalysis of Rijndael // LNCS. 2000. V. 1978. P. 213–230.
22. *Tunstall M.* Improved partial sums-based square attack on AES. Cryptology ePrint Archive, Report 2012/280. 2012.
23. *Leander G., Abdelraheem M. A., AlKhzaimi H., and Zenner E.* A cryptanalysis of PRINTcipher: The invariant subspace attack // CRYPTO 2011. LNCS. 2011. V. 6841. P. 206–221.
24. *Canteaut A., Naya-Plasencia M., and Vayssiere B.* Sieve-in-the-Middle: Improved MITM Attacks (full version). Cryptology ePrint Archive, Report 2013/324. 2013.
25. *Bouillaguet C., Derbez P., Dunkelman O., et al.* Low Data Complexity Attacks on AES. Cryptology ePrint Archive, Report 2010/633. 2010.
26. *Bogdanov A., Chang D., Ghosh M., and Sanadhya S. K.* Bicliques with Minimal Data and Time Complexity for AES (extended version). Cryptology ePrint Archive, Report 2014/932. 2014.
27. *Li L., Jia K., and Wang X.* Improved Meet-in-the-Middle Attacks on AES-192 and PRINCE. Cryptology ePrint Archive, Report 2013/573. 2013.
28. *Gilbert H. and Minier M.* A collision attack on the 7-rounds Rijndael // AES Candidate Conference. N. Y., 2000. P. 230–241.

29. *Demirci H. and Selcuk A.* A meet-in-the-middle attack on 8-round AES // FSE. LNCS. 2008. V. 5086. P. 116–126.
30. *Demirci H., Taskn I., Coban M., and Baysal A.* Improved meet-in-the-middle attacks on AES // INDOCRYPT 2009. LNCS. 2009. V. 5922. P. 144–156.
31. *Xiaoli D., Yupu H., Yongzhuang W., and Jie C.* A new method for meet-in-the-middle attacks on reduced AES // Wireless Communication Over Zigbee for Automotive Inclination Measurement. China Communications. 2011. V. 8. No. 2. P. 21–25.
32. *Wei Y., Lu J., and Hu Y.* Meet-in-the-Middle Attack on 8 Rounds of the AES Block Cipher under 192 Key Bits. Cryptology ePrint Archive, Report 2010/537. 2010.
33. *Bouillaguet C., Derbez P., and Fouque P.-A.* Automatic Search of Attacks on Round-Reduced AES and Applications. Cryptology ePrint Archive, Report 2012/069. 2012.
34. *Derbez P. and Fouque P.-A.* Exhausting Demirci-Selcuk Meet-in-the-Middle Attacks against Reduced-Round AES. Cryptology ePrint Archive, Report 2015/259. 2015.
35. *Biryukov A. and Nikolić I.* Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and others. Cryptology ePrint Archive, Report 2010/248. 2010.
36. *Tunstall M.* Practical Complexity Differential Cryptanalysis and Fault Analysis of AES. Cryptology ePrint Archive, Report 2011/453. 2011.
37. *Cheon J. H., Kim M., Kim K., et al.* Improved impossible differential cryptanalysis of Rijndael and Crypton // ICISC 2001. LCNS. 2002. V. 2288. P. 39–49.
38. *Zhang W., Wu W., and Feng D.* New results on impossible differential cryptanalysis of reduced AES // ICISC 2007. LCNS. 2007. V. 4817. P. 239–250.
39. *Alda F., Aragona R., Nicolodi L., and Sala M.* Implementation and Improvement of the Partial Sum Attack on 6-Round AES. Cryptology ePrint Archive, Report 2014/216. 2014.
40. *Bahrak B. and Aref M.* A novel impossible differential cryptanalysis of AES // Western European Workshop on Research in Cryptology. Bochum, 2007. P. 152-156.
41. *Bahrak B. and Aref M.* Impossible differential attack on seven-round AES-128 // IET Inform. Sec. 2008. V. 2. No. 2. P. 28–32.
42. *Yuan Z.* New Impossible Differential Attacks on AES. Cryptology ePrint Archive, Report 2010/093. 2010.
43. *Dunkelman O., Keller N., and Shamir A.* Improved Single-Key Attacks on 8-round AES. Cryptology ePrint Archive, Report 2010/322. 2010.
44. *Mala H., Dakhilalian M., Rijmen V., and Modarres-Hashemi M.* Improved impossible differential cryptanalysis of 7-round AES-128 // INDOCRYPT 2010. LNCS. 2010. V. 6498. P. 282–291.
45. *Derbez P., Fouque P.-A., and Jean J.* Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. Cryptology ePrint Archive, Report 2012/477. 2012.
46. *Liu Y., Gu D., Liu Z., et al.* New improved impossible differential attack on reduced-round AES-128 // Lecture Notes Electr. Eng. 2012. V. 114. P. 453–461.
47. *Bogdanov A., Kavun E. B., Paar C., et al.* Better than brute-force optimized hardware architecture for efficient biclique attacks on AES-128 // SHARCS12 — Special-Purpose Hardware for Attacking Cryptographic Systems. Washington, 2012. P. 17–34.
48. *Chang D., Ghosh M., and Sanadhya S.* Biclique Cryptanalysis of Full Round AES with Reduced Data Complexity. IIIT Delhi. 2013. `https://repository.iiitd.edu.in/jspui/handle/123456789/99`.

REFERENCES

1. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf — National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, 2001.
2. *Bogdanov A., Khovratovich D., and Rechberger C.* Biclique cryptanalysis of the full AES. ASIACRYPT 2011, LNCS, 2011, vol. 7073, pp. 344–371.
3. *Gilbert H. and Peyrin T.* Super-sbox Cryptanalysis: Improved Attacks for AES-like Permutations. Cryptology ePrint Archive, Report 2009/531, 2009.
4. *Grassi L., Rechberger C., and Ronjom S.* Subspace Trail Cryptanalysis and its Applications to AES. Cryptology ePrint Archive, Report 2016/592, 2016.
5. *Daemen J. and Rijmen V.* The Design of Rijndael: AES — The Advanced Encryption Standard. Berlin, Springer, 2002, 238 p.
6. *Dunkelman O. and Keller N.* The effects of the omission of last round's mixcolumns on AES. Inform. Proc. Let., 2010, vol. 110, no. 8–9, pp. 304–308.
7. *Daemen J. and Rijmen V.* AES Proposal: Rijndael. 1998. http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf.
8. *Bouillaguet C., Derbez P., Dunkelman O., et al.* Low-data complexity attacks on AES. IEEE Trans. Inform. Theory, 2012, vol. 58, no. 11, pp. 7002–7017.
9. *Bulygin S. and Brickenstein M.* Obtaining and solving systems of equations in key variables only for the small variants of AES. Cryptology ePrint Archive, Report 2008/435, 2008.
10. *Van Tilborg H.* Encyclopedia of Cryptography and Security. Berlin, Springer, 2005, 684 p.
11. *Tunstall M.* Practical complexity differential cryptanalysis and fault analysis of AES. J. Cryptographic Eng., 2011, vol. 1, no. 3, pp. 219–230.
12. *Bogdanov A. and Pyshkin A.* Algebraic Side-Channel Collision Attacks on AES. Cryptology ePrint Archive, Report 2007/477, 2007.
13. *Osvik D. A., Shamir A., and Tromer E.* Cache Attacks and Countermeasures: the Case of AES. Cryptology ePrint Archive, Report 2005/271, 2005.
14. *Ali S. S., Mukhopadhyay D., and Tunstall M.* Differential Fault Analysis of AES: Towards Reaching its Limits. Cryptology ePrint Archive, Report 2012/446, 2012.
15. *Biryukov A., Dunkelman O., Keller N., et al.* Key Recovery Attacks of Practical Complexity on AES Variants with up to 10 Rounds. Cryptology ePrint Archive, Report 2009/374, 2009.
16. *Biham E. and Keller N.* Cryptanalysis of reduced variants of Rijndael. Proc. 3rd AES Conf., N. Y., 1999, pp. 11–15.
17. *Knudsen L. R.* Truncated and higher order differentials. LNCS, 1995, vol. 1008, pp. 196–211.
18. *Lu J., Dunkelman O., Keller N., and Kim J.* New Impossible Differential Attacks on AES. Cryptology ePrint Archive, Report 2008/540, 2008.
19. *Tiessen T.* Polytopic cryptanalysis. Proc. 35th Ann. Intern. Conf. Advances in Cryptology — EUROCRYPT 2016, vol. 9665, N. Y., Springer, 2016, pp. 214–239.
20. *Daemen J., Knudsen L., and Rijmen V.* The block cipher square. LNCS, 1997, vol. 1267, pp. 149–165.
21. *Ferguson N., Kelsey J., Lucks S., et al.* Improved cryptanalysis of Rijndael. LNCS, 2000, vol. 1978, pp. 213–230.
22. *Tunstall M.* Improved partial sums-based square attack on AES. Cryptology ePrint Archive, Report 2012/280, 2012.
23. *Leander G., Abdelraheem M. A., AlKhzaimi H., and Zenner E.* A cryptanalysis of PRINTcipher: The invariant subspace attack. CRYPTO 2011, LNCS, 2011, vol. 6841, pp. 206–221.

24. *Canteaut A., Naya-Plasencia M., and Vayssiere B.* Sieve-in-the-Middle: Improved MITM Attacks (full version). Cryptology ePrint Archive, Report 2013/324, 2013.
25. *Bouillaguet C., Derbez P., Dunkelman O., et al.* Low Data Complexity Attacks on AES. Cryptology ePrint Archive, Report 2010/633, 2010.
26. *Bogdanov A., Chang D., Ghosh M., and Sanadhya S. K.* Bicliques with Minimal Data and Time Complexity for AES (extended version). Cryptology ePrint Archive, Report 2014/932, 2014.
27. *Li L., Jia K., and Wang X.* Improved Meet-in-the-Middle Attacks on AES-192 and PRINCE. Cryptology ePrint Archive, Report 2013/573, 2013.
28. *Gilbert H. and Minier M.* A collision attack on the 7-rounds Rijndael. AES Candidate Conference, N. Y., 2000, pp. 230–241.
29. *Demirci H. and Selcuk A.* A meet-in-the-middle attack on 8-round AES. FSE, LNCS, 2008, vol. 5086, pp. 116–126.
30. *Demirci H., Taskn I., Coban M., and Baysal A.* Improved meet-in-the-middle attacks on AES. INDOCRYPT 2009, LNCS, 2009, vol. 5922, pp. 144–156.
31. *Xiaoli D., Yupu H., Yongzhuang W., and Jie C.* A new method for meet-in-the-middle attacks on reduced AES. Wireless Communication Over Zigbee for Automotive Inclination Measurement. China Communications, 2011, vol. 8, no. 2, pp. 21–25.
32. *Wei Y., Lu J., and Hu Y.* Meet-in-the-Middle Attack on 8 Rounds of the AES Block Cipher under 192 Key Bits. Cryptology ePrint Archive, Report 2010/537, 2010.
33. *Bouillaguet C., Derbez P., and Fouque P.-A.* Automatic Search of Attacks on Round-Reduced AES and Applications. Cryptology ePrint Archive, Report 2012/069, 2012.
34. *Derbez P. and Fouque P.-A.* Exhausting Demirci-Selcuk Meet-in-the-Middle Attacks against Reduced-Round AES. Cryptology ePrint Archive, Report 2015/259, 2015.
35. *Biryukov A. and Nikolić I.* Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and others. Cryptology ePrint Archive, Report 2010/248, 2010.
36. *Tunstall M.* Practical Complexity Differential Cryptanalysis and Fault Analysis of AES. Cryptology ePrint Archive, Report 2011/453, 2011.
37. *Cheon J. H., Kim M., Kim K., et al.* Improved impossible differential cryptanalysis of Rijndael and Crypton. ICISC 2001, LCNS, 2002, vol. 2288, pp. 39–49.
38. *Zhang W., Wu W., and Feng D.* New results on impossible differential cryptanalysis of reduced AES. ICISC 2007, LCNS, 2007, vol. 4817, pp. 239–250.
39. *Alda F., Aragona R., Nicolodi L., and Sala M.* Implementation and Improvement of the Partial Sum Attack on 6-Round AES. Cryptology ePrint Archive, Report 2014/216, 2014.
40. *Bahrak B. and Aref M.* A novel impossible differential cryptanalysis of AES. Western European Workshop on Research in Cryptology, Bochum, 2007, pp. 152-156.
41. *Bahrak B. and Aref M.* Impossible differential attack on seven-round AES-128. IET Inform. Sec., 2008, vol. 2, no. 2, pp. 28–32.
42. *Yuan Z.* New Impossible Differential Attacks on AES. Cryptology ePrint Archive, Report 2010/093, 2010.
43. *Dunkelman O., Keller N., and Shamir A.* Improved Single-Key Attacks on 8-round AES. Cryptology ePrint Archive, Report 2010/322, 2010.
44. *Mala H., Dakhilalian M., Rijmen V., and Modarres-Hashemi M.* Improved impossible differential cryptanalysis of 7-round AES-128. INDOCRYPT 2010, LNCS, 2010, vol. 6498, pp. 282–291.
45. *Derbez P., Fouque P.-A., and Jean J.* Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. Cryptology ePrint Archive, Report 2012/477, 2012.

46. *Liu Y., Gu D., Liu Z., et al.* New improved impossible differential attack on reduced-round AES-128. Lecture Notes Electr. Eng., 2012, vol. 114, pp. 453–461.
47. *Bogdanov A., Kavun E. B., Paar C., et al.* Better than brute-force optimized hardware architecture for efficient biclique attacks on AES-128. SHARCS12 — Special-Purpose Hardware for Attacking Cryptographic Systems, Washington, 2012, pp. 17–34.
48. *Chang D., Ghosh M., and Sanadhya S.* Biclique Cryptanalysis of Full Round AES with Reduced Data Complexity. IIIT Delhi, 2013. `https://repository.iiitd.edu.in/jspui/handle/123456789/99`.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.056.53

ИСПОЛЬЗОВАНИЕ ОТЛАДОЧНОГО API СОВРЕМЕННОГО ВЕБ-ОБОЗРЕВАТЕЛЯ ДЛЯ ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ КЛАССА DOM-BASED XSS

Д. А. Сигалов, А. В. Раздобаров, А. А. Петухов

Московский государственный университет имени М. В. Ломоносова, г. Москва, Россия

Рассматривается решение задачи поиска уязвимостей класса DOM-based XSS через последовательную комбинацию методов динамического анализа и fuzz-тестирования. Для создания поддерживаемого динамического анализатора JavaScript-кода используется современный веб-обозреватель Firefox без модификации его исходного кода. Приводится обзор существующих методов поиска уязвимостей класса DOM-based XSS.

Ключевые слова: *DOM-based XSS, уязвимости веб-приложений, динамический анализ, fuzz-тестирование.*

DOI 10.17223/20710410/35/6

DETECTING DOM-BASED XSS VULNERABILITIES USING DEBUG API OF THE MODERN WEB-BROWSER

D. A. Sigalov, A. V. Razdobarov, A. A. Petukhov

Lomonosov Moscow State University, Moscow, Russia

E-mail: asterite@seclab.cs.msu.su, korse@seclab.cs.msu.su, petand@seclab.cs.msu.su

In recent years, a significant part of web application functionality moves to client side. Increasing complexity of client-side code leads to a considerable growth in the number of client-side vulnerabilities and even to an emergence of new types of vulnerabilities, among which DOM-based XSS is most well-known. In this paper, we present an approach to detect and validate DOM-based XSS vulnerabilities. Our approach leverages dynamic tracking of data flows on the client side of web application to identify insecure ones (those which lead to vulnerability). An insecure data flow is a flow, in which a critical operation is data-dependent on attacker-controlled data, which is not sanitised properly. Data flows are tracked and classified using taint propagation technique (also known as "taint checking"). Potentially insecure data flows are tested for the presence of exploitable vulnerability by means of fuzzing — enumeration of possible attack vectors, which are passed to a data flow source. Vulnerability is confirmed if an execution of any injected payload is observed. Our approach was implemented on top of Firefox browser, controlled via it's debugger API. The paper discusses and justifies advantages of such implementation. The paper also provides an analysis of related work in the subject field, and comparison with other approaches is made. The proposed approach and its implementation are maintainable and extensible, which is

crucial for analyzing applications in constantly evolving environment such as client side web technologies.

Keywords: *DOM-based XSS, web application vulnerabilities, dynamic analysis, fuzzing.*

Введение

В работе предлагается решение задачи автоматизированного поиска уязвимостей типа DOM-based XSS (далее — DOMXSS). Уязвимости DOMXSS являются частными случаями уязвимостей к атакам Cross Site Scripting (XSS) — уязвимостей веб-приложений, которые предоставляют возможность злоумышленникам передать и выполнить подготовленный ими JavaScript-код в веб-обозревателе легитимного пользователя веб-приложения от его имени. Специфика DOMXSS заключается в том, что причиной уязвимости является ошибка в коде клиентской стороны веб-приложения: в JavaScript-коде, который выполняется на веб-странице и реализует интерактивный интерфейс взаимодействия с пользователем.

Более формально, уязвимость к атаке XSS имеет место, когда веб-приложение выводит данные, полученные из недоверенного источника (например, от пользователя), в HTTP-ответ таким образом, что они либо формируют новый контекст выполнения JavaScript-кода на странице, визуализируемой веб-обозревателем, либо нарушают целостность существующего контекста выполнения JavaScript-кода. Далее такой вывод данных в HTTP-ответ будем называть «влиянием на контекст выполнения JavaScript-кода» или просто «влиянием на контекст выполнения».

В случае DOMXSS недоверенным источником данных является окружение JavaScript-кода, на которое может влиять потенциальный атакующий: URL веб-страницы, элементы веб-страницы, значения которых формируются на основе ввода пользователей веб-приложения, и т. д. Влияние на контекст выполнения JavaScript-кода может произойти из-за небезопасного использования разработчиками кода веб-страницы, методов и свойств интерфейса объектной модели веб-страницы (DOM API). К таким методам относятся вызовы, предназначенные для модификации содержимого веб-страницы в интерактивном режиме (например, метод `document.write`), или вызовы, предназначенные для выполнения JavaScript-кода на лету (например, метод `eval`). Полный перечень небезопасных методов и свойств DOM API приведён в документе DOM XSS Wiki [1]. Недоверенные данные, используемые в подобных API-вызовах, открывают возможность злоумышленнику выполнить в контексте веб-страницы жертвы атаки произвольный JavaScript-код.

Пример уязвимости рассматриваемого типа содержится в веб-странице, фрагмент JavaScript-кода которой приведен в листинге 1. В JavaScript-коде продемонстрирован небезопасный вывод URL-адреса веб-страницы в её заголовок.

```
1 <html><body>
2       <h1>Поделиться адресом страницы: </h1>
3       <script>document.querySelector('h1').innerHTML +=
    location.href; </script>
4 </body></html>
```

Листинг 1. Пример веб-страницы, содержащей уязвимость DOMXSS

Если злоумышленник вынудит жертву атаки перейти по ссылке с адресом `http://site.com/page.html#<img src=x onerror=alert('XSS');></img>`, то после выпол-

нения операции `document.querySelector('h1').innerHTML += location.href` структура документа станет следующей:

```
1 <html><body>
2       <h1>Поделиться адресом страницы: http://site.com/page
    .html#<img src=x onerror=alert('XSS');></img></h1>
3       <script>document.querySelector('h1').innerHTML +=
    location.href; </script>
4 </body></html>
```

Листинг 2. Пример веб-страницы, содержащей уязвимость DOMXSS, после её эксплуатации

В результате изменения структуры веб-страницы появился новый контекст выполнения JavaScript-кода, подконтрольный злоумышленнику, который в приведённом примере содержит вызов «`alert('XSS')`». Выполнение внедрённого JavaScript-кода в контексте веб-страницы приводит к появлению окна-оповещения с надписью «XSS», что продемонстрировано на снимке экрана (рис. 1), демонстрация проведена в веб-обозревателе Google Chrome.

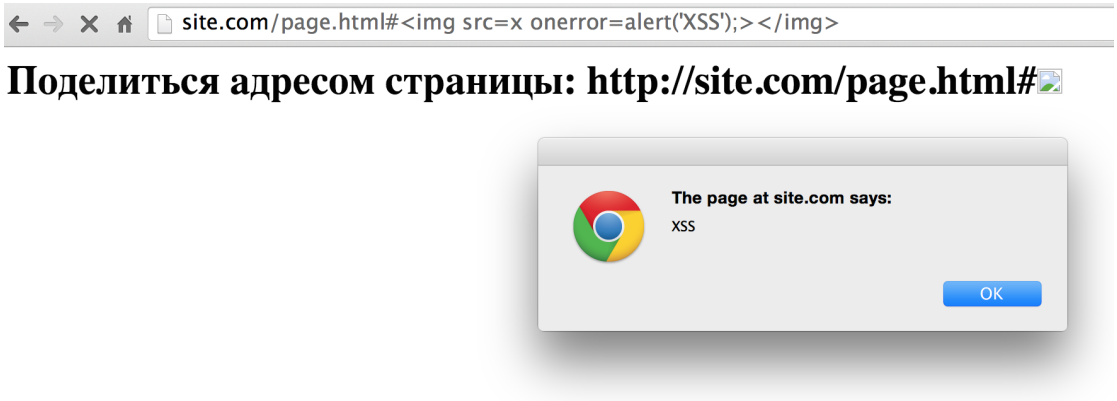

Рис. 1. Результат выполнения JavaScript-кода, внедрённого в веб-страницу с кодом из листинга 1

Добиться перехода по ссылке, что приведёт к эксплуатации уязвимости, можно, например, с помощью веб-страницы со следующим кодом (листинг 3); это проверено на нескольких версиях браузера Google Chrome, включая 55.0.2883.87 m.

```
1 <html><body>
2 Free movies: <a href="http://site.com/page.html#<img src=x
    onerror=alert('XSS');></img>">Click here for free
    downloads!</a>
3 </body></html>
```

Листинг 3. Пример веб-страницы, используемой для эксплуатации уязвимости

Для обнаружения уязвимостей типа DOMXSS необходимо проверить, используются ли данные из недоверенного источника в небезопасных методах DOM API (модификация содержимого веб-страницы или выполнение кода на лету) без предварительной обработки. Данный факт может быть установлен различными способами:

— с помощью fuzz-тестирования веб-страницы, её свойств и способов пользовательского взаимодействия с ней с последующим отслеживанием факта выполнения внед-

рённого JavaScript-кода (например, можно отслеживать вызов функции alert с заданным аргументом);
— с помощью статического или динамического анализа JavaScript-кода.

Каждый из приведённых способов обладает существенными ограничениями. Статический анализ JavaScript-кода является трудноразрешимой задачей в силу следующих обстоятельств:

— из-за динамической типизации языка;
— из-за возможности генерации и исполнения JavaScript-кода на лету (см. функцию `eval`);
— за счёт существенной зависимости логики выполнения JavaScript-кода от свойств DOM веб-страницы, значения которых определяются только на этапе работы пользователя с веб-приложением (данные свойства в общем случае зависят и от состояния веб-приложения на стороне сервера).

Реализация методов динамического анализа сопряжена с необходимостью тесной интеграции с интерпретатором языка JavaScript. Поскольку подавляющее большинство интерпретаторов не предоставляет вовне API для гранулярного контроля своей работы, часто единственным выходом для реализации динамического анализа остаётся модификация кода самого интерпретатора. Авторам не известна ни одна реализация динамического анализатора, который поддерживался бы его создателями параллельно с развитием основного интерпретатора. С учётом динамики появления новых версий стандарта ECMAScript [2] описанное ограничение является существенным.

Реализация fuzz-тестирования сопряжена с большим пространством перебора: учитывая, что данные из недоверенного источника могут небезопасно записываться в свойства DOM в результате самой сложной последовательности действий пользователя в веб-интерфейсе страницы, fuzz-тестирование без эффективной стратегии выбора некорректных входных данных приведёт к низкой полноте анализа. Сопутствующей технической сложностью является необходимость восстанавливать предыдущее состояние веб-страницы при последовательном переборе различных цепочек пользовательских действий.

В данной работе предлагается подход, который позволяет создать поддерживаемое, стабильно работающее средство, сочетающее преимущества динамического анализа JavaScript-кода и fuzz-тестирования, но лишённое их основных недостатков. Предлагаемый подход объединяет в себе следующие техники анализа JavaScript-кода:

— для построения зависимостей по данным во время выполнения JavaScript-кода используется метод распространения метки (taint-анализ) [3, 4];
— для обеспечения поддерживаемости реализации динамического taint-анализа используется Mozilla Debugger API веб-обозревателя Mozilla Firefox [5], позволяющий необходимым образом контролировать выполнение JavaScript-кода;
— для подтверждения небезопасности найденных потоков данных в части возможности проведения атаки класса DOMXSS используется метод направленного fuzz-тестирования, разработанный авторами. Направленное fuzz-тестирование позволяет существенно сократить пространство полного перебора за счёт использования информации о специфике обработки входных данных в тестируемом фрагменте JavaScript-кода, которая собирается на этапе динамического анализа.

1. Существующие методы и средства

Уязвимость типа XSS является примером уязвимости, описываемой моделью невмешательства [6]. Эта модель подразумевает наличие в программе или её фрагменте доверенного и недоверенного каналов ввода данных, а также доверенного и недоверенного каналов вывода данных. Под каналом вывода, в общем случае, понимаются вызовы подпрограмм (функций, процедур, методов) с аргументами или присваивания данных переменным или полям структур и объектов. Наличие уязвимости устанавливается как передача данных из недоверенного канала ввода в доверенный канал вывода. Модель невмешательства удобно описывает типы уязвимостей, которые появляются в результате использования программистами недоверенных данных, поступивших от пользователя, в критичных API-вызовах. Уязвимости типа DOMXSS являются частным случаем таких уязвимостей. Для применения модели невмешательства для поиска уязвимостей класса DOMXSS достаточно недоверенным каналам ввода поставить в соответствие методы и свойства DOM-модели, позволяющие JavaScript-коду получить данные, контролируемые пользователем, а доверенным каналам вывода — методы и свойства DOM-модели, позволяющие изменять структуру веб-страницы или выполнять JavaScript-код на лету.

1.1. Статический анализ

В существующих работах установление факта небезопасной передачи данных без выполнения анализируемого кода рассматривается либо как задача вычисления потоков данных от одних операторов (канал ввода) до других (канал вывода), либо как задача поиска небезопасной последовательности операторов над различными представлениями программы (дерево синтаксического разбора, граф потоков управления), либо как задача определения множества возможных значений переменной в заданной точке программы (в канале вывода).

Достоинством статического анализа является сбор информации обо всём имеющемся коде, вне зависимости от того, выполнялся он или нет. В контексте анализа кода на языке JavaScript, однако, применение статического анализа затруднительно. Язык предоставляет возможность динамически формировать или загружать с сервера код, а затем выполнять его. Данное обстоятельство порождает большую неопределённость при статическом анализе. Как следствие, существующие методы [7 – 10] накладывают существенные ограничения на анализируемый код — например, требуют отсутствия динамически вычисляемых имён свойств, неизменности прототипов объектов и самих объектов после первичной инициализации, отсутствия динамического формирования и выполнения кода и т. п. При вычислении множества возможных значений переменных в заданной точке JavaScript-программы сложность вызывают многочисленные обращения к свойствам DOM-элементов, значения которых становятся известны только во время выполнения. Точность аппроксимации значений, получаемых из страниц или Ajax-запросов, напрямую определяет качество итогового анализа.

Перечисленные характеристики программ на языке JavaScript позволяют сделать вывод о неприменимости статического анализа для поиска уязвимостей типа DOMXSS без расширения его методами динамического анализа.

1.2. Динамический анализ

Известны два подхода для динамического анализа JavaScript-кода.

Первый подход заключается в создании динамического анализатора на языке JavaScript. Анализ JavaScript-программы из другой JavaScript-программы можно осуществить благодаря различным методам интроспекции, которые поддерживаются

стандартом языка. При этом JavaScript-код анализатора встраивается в веб-страницу и контролирует выполнение анализируемого JavaScript-кода с помощью средств (в том числе методами интроспекции), которые предоставляет интерпретатор веб-обозревателя и DOM API. Основные методы, используемые при данном подходе, — это инструментирование анализируемого кода собственными вызовами, а также переопределение стандартных функций и объектов собственными. Последний метод позволяет осуществить вызовы к стандартному API через код собственных наблюдателей (т. н. функций-перехватчиков) и отслеживать передаваемые в них аргументы. Преимущество данного подхода заключается в том, что для его реализации не требуется модифицировать интерпретатор JavaScript-кода. Ограничения подхода связаны с уровнем рассмотрения процесса выполнения: наблюдателю на уровне JavaScript-кода недоступна информация о процессе выполнения программы, которая есть на уровне внутренних структур данных интерпретатора языка JavaScript. В результате описанный подход не позволяет реализовать полноценное отслеживание зависимостей по данным.

В то же время этот подход позволяет существенно повысить эффективность fuzz-тестирования веб-страницы при поиске уязвимостей типа DOMXSS. Динамический анализатор симулирует различные пользовательские действия на веб-странице, порождая тем самым DOM-события. DOM-события вызывают выполнение JavaScript-обработчиков с теми или иными значениями, полученными из свойств DOM-элементов. Инструментирование методов получения свойств DOM-элементов, а также методов вывода данных в DOM позволяет реализовать стратегии более эффективного fuzz-тестирования по сравнению с передачей во все входные каналы предустановленного списка тестирующих значений, таких, как `<script>alert(1);</script>`.

Второй подход заключается в создании динамического анализатора, который работает на уровне внутренних структур данных интерпретатора языка. В этом случае наблюдателю доступна любая информация о ходе выполнения кода, что позволяет корректно реализовать динамический taint-анализ и с его помощью устанавливать факт передачи данных из недоверенного канала ввода в доверенный канал вывода. Данный подход технически сложен в реализации, так как требуется среда, удовлетворяющая следующим требованиям:

— наличие интерпретатора JavaScript-кода актуальных версий стандарта ECMAScript;
— среда должна полноценно реализовывать DOM API в объёме текущих спецификаций, которые постоянно обновляются;
— среда должна предоставлять наблюдателю возможность получать информацию о процессе выполнения кода на уровне внутренних структур данных.

Попытки реализовать такую среду через модификацию кода существующих веб-обозревателей (WebKit, Chromium) или с нуля (HtmlUnit [11]) сталкиваются с трудностями при последующей поддержке полученного решения. Действительно, реализация в веб-обозревателях новых функций DOM API или нового стандарта ECMAScript приведёт к тому, что JavaScript-код, использующий новые возможности веб-обозревателей, перестанет выполняться в модифицированных средах для динамического анализа.

Стоит отдельно отметить, что сам метод динамического taint-анализа не лишён ограничений:

— наличие в коде ограничений на входные данные (множество допустимых значений, длина) может сделать код безопасным даже без явного использования фильтрующих функций при передаче данных из недоверенных каналов ввода в доверенные каналы вывода;

— корректность собственных валидирующих функций невозможно установить с помощью динамического taint-анализа (например, если валидация реализована с помощью регулярных выражений).

Приведённые рассуждения позволяют сделать следующие выводы:

— реализация динамического taint-анализа имеет смысл только в среде, поддержка которой осуществляется сообществом;
— отдельной задачей как при реализации динамического taint-анализа, так и при реализации fuzz-тестирования является максимизация покрытия анализируемого кода, что сопряжено с задачей симуляции на веб-странице возможных последовательностей пользовательских действий;
— динамический taint-анализ и fuzz-тестирование эффективно дополняют друг друга при решении задачи поиска уязвимостей типа DOMXSS.

1.3. Существующие средства

В результате исследования предметной области были проанализированы существующие инструментальные средства, решающие задачу поиска уязвимостей типа DOMXSS.

Средства Codemagi Burp DOM-XSS Scanner [12] и StaticBurp [13] используют поиск мест, содержащих уязвимость, по регулярным выражениям по узлам дерева разбора JavaScript-кода (AST). Предполагается, что результаты поиска проверяются оператором средства вручную.

DOMinator [14] использует динамический анализ потоков данных с помощью taint-анализа. Для сбора информации о выполнении используется изменённый веб-обозреватель Firefox под управлением оператора. В задачи оператора входит навигация по веб-приложению с целью максимизации покрытия анализируемого JavaScript-кода.

В средствах, описанных в [15 – 17], также реализован динамический taint-анализ, информация для которого собирается с помощью изменённого кода интерпретатора веб-обозревателя Google Chrome.

Авторы [18 – 20] применяют динамический taint-анализ, собирая информацию о выполнении с помощью преобразований JavaScript-кода.

Инструмент ra2-dom-xss-scanner [21] осуществляет простое fuzz-тестирование веб-страницы и отслеживание изменений свойств DOM.

Инструмент [22], а также авторы [23, 24] используют статический поиск зависимостей по данным между источниками ввода небезопасных данных и их использованием в небезопасных методах и свойствах DOM API.

В [7, 25] применяется комбинация статического и динамического анализа. Статическому анализатору на вход подаётся уже имеющийся на странице JavaScript-код. Если во время работы статического анализа обнаруживается вызов функции динамического выполнения кода, то её аргумент, содержащий JavaScript-код, подлежащий интерпретации на лету, также подаётся на вход статическому анализатору. Кроме того, в средстве [25], если во время статического анализа в коде страницы используются значения, получаемые из окружения кода (например, свойства DOM), они предоставляются средой выполнения в явном виде.

2. Предлагаемый метод

В данной работе предлагается производить динамический taint-анализ JavaScript-кода для поиска небезопасных потоков данных, а затем подтверждать наличие уязвимости в них через направленное fuzz-тестирование.

В качестве среды для анализа выполнения кода выбран веб-обозреватель Firefox. Использование данного веб-обозревателя позволяет максимально приблизить среду анализа к реальным условиям выполнения анализируемого кода и, как следствие, обеспечить корректную работу JavaScript-кода современных веб-приложений. Использование Mozilla Debugger API позволяет избежать модификации кода веб-обозревателя и, таким образом, обеспечить сопровождаемость реализованного метода и в последующих версиях веб-обозревателя.

После обнаружения потенциально небезопасных потоков данных для каждого из них производится попытка подтвердить наличие уязвимости через fuzz-тестирование значений, поступающих в потенциально небезопасный поток через канал ввода. Конкретный набор передаваемых в канал данных зависит от того, какие именно методы DOM API представляют канал ввода и канал вывода. Такой подход позволяет сократить пространство перебора, так как вместо всех возможных значений векторов атаки на вход передаются только те, которые могут привести к модификации или порождению нового контекста выполнения JavaScript-кода в данном конкретном случае.

Описываемый метод не включает решение задачи максимизации покрытия анализируемого JavaScript-кода. Он реализован в виде инструментального средства, которое представляет собой расширение (плагин) для веб-обозревателя Firefox. Предполагается, что оператор осуществляет навигацию по веб-приложению, используя штатные средства веб-обозревателя, а расширение в фоновом режиме вычисляет небезопасные потоки данных. При обнаружении таковых оператору отправляется сигнал, при получении которого он может инициировать этап fuzz-тестирования. В случае успеха последнего оператору выводится сообщение о найденной уязвимости и предоставляются проверочные данные, её подтверждающие.

Далее перечислены детали реализации динамического taint-анализа и fuzz-тестирования.

2.1. Реализация динамического анализа

Динамический taint-анализ является одним из способов определить зависимости по данным в программе в процессе её выполнения. В настоящей работе в процессе динамического taint-анализа участвуют данные только строкового типа (прототип string). В качестве источников, вывод которых устанавливает строкам флаг tainted, берутся все потенциально контролируемые злоумышленником методы и свойства DOM API, полный список которых может быть найден в энциклопедии DOM XSS [1]. Операциями, распространяющими флаг tainted, считаются любые функции и операторы преобразования строк (конкатенация, взятие подстроки, URL-декодирование и т. п.).

Для реализации отслеживания потоков данных используется отладочный интерфейс интерпретатора JavaScript веб-обозревателя Firefox (Mozilla Debugger API [5]). Для хранения информации о флагах заводится специальная таблица, хранящая ссылки на все строки, помеченные флагом tainted.

Более конкретно, taint-анализ реализован следующим образом:

1) Объекты, отвечающие за источники (например, объект `location`, предоставляющий интерфейс для работы с адресом страницы), заменяются средствами отладчика на специальные объекты-обёртки (proxy-объекты). Обёртки ведут себя так же, как и реальные объекты, однако при считывании из них данных (для `location` это произойдёт, например, при приведении к строке методом `toString` или при обращении к полю `location.href`) соответствующая операция запишет в таблицу строк, помеченных

флагом tainted, запрашиваемое значение прежде, чем возвратить его. Кроме самого значения, в таблицу помещается информация о его источнике (канале ввода).

2) Строковые операции, встроенные в язык, также подменяются на прокси-методы. Подменённые операции, кроме своего основного действия, проверяют, есть ли в таблице строки — аргументы операции. Если это так, то результат операции также помещается в таблицу ссылок на taint-строки. В метку источника у новой записи в таблицу заносятся источники всех помеченных аргументов операции.

3) Доверенные каналы вывода представляют функции и свойства выполнения кода на лету и изменения DOM: `eval`, `document.write`, `setTimeout`, `element.innerHTML`, `script.src`. Все такие функции и свойства заменяются на прокси-методы и свойства, которые перед своим основным действием проверяют, не содержится ли передаваемое им значение в таблице. Если это так, то сообщается об обнаружении потенциально небезопасного потока данных.

2.2. Реализация fuzz-тестирования

Цель этапа fuzz-тестирования для всех потенциально небезопасных потоков данных, найденных на этапе динамического taint-анализа, — подобрать такие входные данные, попадание которых в выходной канал может привести к созданию нового контекста выполнения JavaScript-кода или к модификации существующего. Оракулом такого события является вызов функции `alert` с уникальным случайным числом, которое генерируется для каждого экземпляра проверочных входных данных. В случае положительного ответа оракула средство выводит в журнал информацию о протестированном потоке данных и входные данные, подтверждающие небезопасность этого потока.

Так как потенциально небезопасный поток данных может быть обнаружен в результате сложной комбинации пользовательских действий, выполняемых оператором (например, раскрытие выпадающего меню, выбор в нём определённого пункта, переход в появившемся интерфейсе в определённую вкладку и раскрытие выпадающего меню в ней), актуальной задачей является реализация fuzz-тестирования кода таким образом, чтобы оператору для каждого экземпляра тестовых данных не приходилось повторять действия в веб-интерфейсе, активирующие соответствующий поток в JavaScript-коде страницы.

Более конкретно, повторное выполнение участка JavaScript-кода, который активируется пользовательскими действиями в веб-интерфейсе, может быть осуществлено следующим образом:

1) Перед выполнением кода, активированного DOM-событием, полностью сохраняется контекст выполнения JavaScript-кода. Для выполняемого кода оценивается его безопасность с помощью taint-анализа. В случае, если в результате применения taint-анализа поток помечается как потенциально небезопасный, реализуются следующие шаги с целью подтверждения данного факта.

2) В зависимости от типа API-функции, представляющей доверенный выходной канал, генерируются экземпляры входных данных, которые становятся кандидатами для проверки.

3) Для каждого сгенерированного экземпляра входных данных происходит восстановление контекста выполнения JavaScript-кода и управление передаётся на оператор, соответствующий началу тестируемого потока данных.

4) При достижении в процессе выполнения кода оператора, который соответствует получению пользовательских данных, в качестве результата его вызова возвращается очередной экземпляр тестовых входных данных.

5) Выполнение продолжается до тех пор, пока не выполнится последний оператор, представленный в тестируемом потоке данных, после чего осуществляется переход на шаг 3, перед чем осуществляется восстановление контекста выполнения (см. шаг 1).

Выполнение функции `alert` при достижении канала вывода очередных тестовых данных может происходить как синхронно, так и асинхронно. В первом случае процесс fuzz-тестирования для текущего потока завершается и информация о подтверждённой уязвимости записывается в журнал.

С учётом того, что выполнение функции `alert` может произойти асинхронно, возможно даже после выполнения оператором в веб-интерфейсе приложения каких-то других действий, факт вызова функции `alert` должен отслеживаться на всём протяжении работы средства. При этом уникальный идентификатор, являющийся аргументом функции `alert`, однозначно определит последовательность действий пользователя в интерфейсе приложения и уязвимые фрагменты кода, приводящие к порождению асинхронного контекста выполнения JavaScript-кода с функцией `alert`.

Описанный метод может быть реализован при достаточно большой степени контроля над выполнением кода. Такой контроль предоставляется программным интерфейсом отладчика Firefox, который и используется для реализации анализа процесса выполнения кода в данной работе.

Заключение

Предложен метод поиска уязвимостей типа DOMXSS с помощью динамического taint-анализа и последующего направленного fuzz-тестирования. Метод реализован на основе веб-обозревателя Firefox без модификации его исходного кода. Динамический taint-анализ позволяет обнаруживать потенциально небезопасные места в коде, в том числе динамически сформированном или загруженном на страницу в результате асинхронных запросов. Использование направленного fuzz-тестирования позволяет избавиться от ложных срабатываний, возникающих из-за ограничений динамического taint-анализа, а также получать на выходе проверочные данные, подтверждающие наличие уязвимостей. Использование в качестве среды для динамического анализа немодифицированного веб-обозревателя Firefox позволяет выполнять анализ кода в реальном окружении и избавляет от необходимости осуществлять постоянную поддержку средства при выходе обновлений веб-обозревателя.

ЛИТЕРАТУРА

1. *Stefano D. P., Heiderich M., and Braun F.* `https://code.google.com/p/domxsswiki/` — DOM XSS Test Cases Wiki Cheatsheet Project. 2010.
2. `http://www.ecma-international.org/publications/standards/Ecma-262.htm` — Стандарты языка программирования ECMAScript.
3. *Schwartz E. J., Avgerinos T., and Brumley D.* All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask) // IEEE Symp. Security and Privacy. Auckland: IEEE, 2010. P. 317–331.
4. *Wagner D.* Static analysis and software assurance. SAS'01. Proc. 8th Intern. Symp. Static Analysis. Paris: Springer Verlag, 2001. 431 p.
5. `https://developer.mozilla.org/en-US/docs/Tools/Debugger-API` — Документация отладочного интерфейса Mozilla Debugger API.

6. *McLean J.* Security Models // Encyclopedia of Software Engineering. N. Y.: John Wiley & Sons Inc., 1994. P. 1136–1145.
7. *Chugh R. et al.* Staged information flow for JavaScript // ACM Sigplan Notices. 2009. V. 44. No. 6. P. 50–52.
8. *Guha A., Krishnamurthi S., and Jim T.* Using static analysis for Ajax intrusion detection // Proc. 18th Intern. Conf. on World Wide Web. Madrid: ACM, 2009. P. 561–570.
9. *Madsen M., Livshits B., and Fanning M.* Practical static analysis of JavaScript applications in the presence of frameworks and libraries // Proc. 9th Joint Meeting on Foundations of Software Engineering. St. Petersburg: ACM, 2013. P. 499–509.
10. *Feldthaus A. et al.* Efficient construction of approximate call graphs for JavaScript IDE services // 35th Intern. Conf. Software Engineering (ICSE). San Francisco: IEEE, 2013. P. 752–761.
11. `http://htmlunit.sourceforge.net/` — Документация эмулятора управляемого веб-обозревателя HtmlUnit. 2016.
12. `https://www.codemagi.com/downloads/dom-xss-scanner-checks` — Описание Codemagi Burp DOM-XSS Scanner — расширения для инструмента BurpSuite, предназначенного для поиска DOMXSS.
13. `http://www.ethicalhack3r.co.uk/staticburp-burp-suite-potential-dom-xss-analysis/` — Описание StaticBurp — расширения для инструмента BurpSuite, предназначенного для поиска DOMXSS.
14. `https://dominator.mindedsecurity.com/` — Описание средства поиска DOMXSS DOMinator
15. *Just S. et al.* Information flow analysis for JavaScript // Proc. 1st ACM SIGPLAN Intern. Workshop on Programming Language and Systems Technologies. Portland: ACM, 2011. P. 9–18.
16. *Lekies S., Stock B., and Johns M.* 25 million flows later: large-scale detection of DOM-based XSS // Proc. ACM SIGSAC Conf. Computer & Communications Security. Berlin: ACM, 2013. P. 1193–1204.
17. *Lekies S., Stock B., and Johns M.* Practical blended taint analysis for JavaScript // Proc. Intern. Symp. Software Testing and Analysis. Lugano: ACM, 2013. P. 336–346.
18. *Xiao W. et al.* Preventing client side XSS with rewrite based dynamic information flow // Sixth Intern. Symp. Parallel Architectures, Algorithms and Programming (PAAP). Pekin: IEEE, 2014. P. 238–243.
19. *Jang D. et al.* Rewriting-based dynamic information flow for JavaScript // 17th ACM Conf. Computer and Communications Security. Chicago: ACM, 2010. `http://goto.ucsd.edu/~rjhala/papers/rewriting_based_dynamic_information_flow_for_javascript.pdf`
20. *Prabawa A. and Chin W. N.* Titania: Generic Dynamic Information Flow Analysis Framework for JavaScript. `http://www.comp.nus.edu.sg/~adi-yoga/Titania/Titania.pdf` (дата обращения: 01.09.2016).
21. `https://code.google.com/p/ra2-dom-xss-scanner/` — Документация средства поиска DOMXSS Ra.2.
22. `www.jsprime.org` — Описание средства поиска DOMXSS JSPrime.
23. *Guarnieri S. et al.* Saving the world wide web from vulnerable JavaScript // Proc. Intern. Symp. Software Testing and Analysis. Toronto: ACM, 2011. P. 177–187.
24. *Saxena P. et al.* A symbolic execution framework for JavaScript // IEEE Symp. Security and Privacy. Auckland: IEEE, 2010. P. 513–528.

25. *Tripp O., Ferrara P., and Pistoia M.* Hybrid security analysis of Web JavaScript code via dynamic partial evaluation // Proc. Intern. Symp. Software Testing and Analysis. San Jose: ACM, 2014. P. 49–59.

REFERENCES

1. *Stefano D. P., Heiderich M., and Braun F.* https://code.google.com/p/domxsswiki/ — DOM XSS Test Cases Wiki Cheatsheet Project, 2010.
2. http://www.ecma-international.org/publications/standards/Ecma-262.htm
3. *Schwartz E. J., Avgerinos T., and Brumley D.* All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask). IEEE Symp. Security and Privacy, Auckland, IEEE, 2010, pp. 317–331.
4. *Wagner D.* Static analysis and software assurance. SAS'01. Proc. 8th Intern. Symp. Static Analysis, Paris, Springer Verlag, 2001. 431 p.
5. https://developer.mozilla.org/en-US/docs/Tools/Debugger-API
6. *McLean J.* Security Models. Encyclopedia of Software Engineering, N. Y., John Wiley & Sons Inc., 1994, pp. 1136–1145.
7. *Chugh R. et al.* Staged information flow for JavaScript. ACM Sigplan Notices, 2009, vol. 44, no. 6, pp. 50–52.
8. *Guha A., Krishnamurthi S., and Jim T.* Using static analysis for Ajax intrusion detection. Proc. 18th Intern. Conf. on World Wide Web, Madrid, ACM, 2009, pp. 561–570.
9. *Madsen M., Livshits B., and Fanning M.* Practical static analysis of JavaScript applications in the presence of frameworks and libraries. Proc. 9th Joint Meeting on Foundations of Software Engineering, St. Petersburg, ACM, 2013, pp. 499–509.
10. *Feldthaus A. et al.* Efficient construction of approximate call graphs for JavaScript IDE services. 35th Intern. Conf. Software Engineering (ICSE), San Francisco, IEEE, 2013, pp. 752–761.
11. http://htmlunit.sourceforge.net/
12. https://www.codemagi.com/downloads/dom-xss-scanner-checks
13. http://www.ethicalhack3r.co.uk/staticburp-burp-suite-potential-dom-xss-analysis/
14. https://dominator.mindedsecurity.com/ —
15. *Just S. et al.* Information flow analysis for JavaScript. Proc. 1st ACM SIGPLAN Intern. Workshop on Programming Language and Systems Technologies, Portland, ACM, 2011, pp. 9–18.
16. *Lekies S., Stock B., and Johns M.* 25 million flows later: large-scale detection of DOM-based XSS. Proc. ACM SIGSAC Conf. Computer & Communications Security, Berlin, ACM, 2013, pp. 1193–1204.
17. *Lekies S., Stock B., and Johns M.* Practical blended taint analysis for JavaScript. Proc. Intern. Symp. Software Testing and Analysis, Lugano, ACM, 2013, pp. 336–346.
18. *Xiao W. et al.* Preventing client side XSS with rewrite based dynamic information flow. Sixth Intern. Symp. Parallel Architectures, Algorithms and Programming (PAAP), Pekin, IEEE, 2014, pp. 238–243.
19. *Jang D. et al.* Rewriting-based dynamic information flow for JavaScript. 17th ACM Conf. Computer and Communications Security, Chicago, ACM, 2010. http://goto.ucsd.edu/~rjhala/papers/rewriting_based_dynamic_information_flow_for_javascript.pdf
20. *Prabawa A. and Chin W. N.* Titania: Generic Dynamic Information Flow Analysis Framework for JavaScript. http://www.comp.nus.edu.sg/~adi-yoga/Titania/Titania.pdf (access date 01.09.2016).
21. https://code.google.com/p/ra2-dom-xss-scanner/

22. `www.jsprime.org`
23. *Guarnieri S. et al.* Saving the world wide web from vulnerable JavaScript. Proc. Intern. Symp. Software Testing and Analysis, Toronto, ACM, 2011, pp. 177–187.
24. *Saxena P. et al.* A symbolic execution framework for JavaScript. IEEE Symp. Security and Privacy, Auckland, IEEE, 2010, pp. 513–528.
25. *Tripp O., Ferrara P., and Pistoia M.* Hybrid security analysis of Web JavaScript code via dynamic partial evaluation. Proc. Intern. Symp. Software Testing and Analysis, San Jose, ACM, 2014, pp. 49–59.

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

УДК 621.391.7

ПРИМЕНЕНИЕ ОДНОГО МЕТОДА РАСПОЗНАВАНИЯ ЛИНЕЙНОГО КОДА ДЛЯ КАНАЛА С ПОДСЛУШИВАНИЕМ

Ю. В. Косолапов, О. Ю. Турченко

Южный федеральный университет, г. Ростов-на-Дону, Россия

Рассматривается модель защиты данных с помощью метода кодового зашумления. Предполагается, что кодируемые информационные блоки длины k содержат фиксированное сообщение $\mathbf{m}$ длины m_l ($m_l \leqslant k$) на фиксированной позиции m_s ($1 \leqslant m_s \leqslant k - m_l + 1$), а наблюдатель получает зашумлённые кодовые слова длины n через q-ичный (q — степень простого числа) симметричный канал с вероятностью p для каждого ненулевого значения ошибки. Целью наблюдателя является нахождение сообщения $\mathbf{m}$, когда позиция m_s и длина m_l неизвестны. Предложен способ нахождения сообщения $\mathbf{m}$ и получена оценка количества наблюдаемых кодовых слов, достаточного для восстановления сообщения $\mathbf{m}$ этим способом.

Ключевые слова: *кодовое зашумление, q-ичный симметричный канал, распознавание кода.*

DOI 10.17223/20710410/35/7

APPLICATION OF ONE METHOD OF LINEAR CODE RECOGNITION TO THE WIRE-TAP CHANNEL

Y. V. Kosolapov, O. Y. Turchenko

Southern Federal University, Rostov-on-Don, Russia

E-mail: itaim@mail.ru

A security model using the method of code noising is considered. It is assumed that the information blocks of length k contain a fixed message $\mathbf{m}$ of length m_l ($m_l \leqslant k$) on a fixed position m_s ($1 \leqslant m_s \leqslant k - m_l + 1$), and an observer gets noisy codewords of length n through a q-ary symmetric channel $q\mathrm{SC}(p)$ (q — prime power) with error probability p ($p < 1/q$) for each non-zero value. The aim of the observer is to find the unknown message $\mathbf{m}$, when the position m_s and the length m_l are unknown. In this paper, we propose a statistical method for finding message $\mathbf{m}$. The method is based on the idea of code recognition in noisy environment: we test hypothesis H_0 (received vectors have been generated by a conjectural code $\mathcal{C}$) against the hypothesis H_1 (received vectors have not been generated by $\mathcal{C}$ or it's subcode). The method is as follows. From the observed vectors, the independent pairwise differences of them are compiled. The resulting vectors are code words of some unknown code $\mathcal{C}$ noised in a symmetric channel $q\mathrm{SC}(p(2 - qp))$. To determine the length m_l and place m_s of the unknown message $\mathbf{m}$, we recognize the code $\mathcal{C}$ from the calculated differences of the received vectors. For this purpose, we present a code recognition algorithm (called CodeRecognition) and prove that if $\mathcal{C}$ is a linear $[n, k]$ code and $M(\mathcal{C}, \alpha, \beta)$ is

the minimum number of vectors (received from the channel $q\mathrm{SC}(p)$) that are sufficient for CodeRecognition algorithm to test the hypothesis H_0 against the hypothesis H_1 by using a constructed statistical criterion, then $M(\mathcal{C}, \alpha, \beta) \leqslant f^2(k+1)$, where

$$f(x) = \frac{b(1+(q-2)(1-pq)^x - (q-1)(1-pq)^{2x})^{1/2} - a}{\sqrt{q-1}(1-pq)^x},$$

α and β are the probabilities of errors of the first kind and the second kind, respectively; $a = \Phi^{-1}(\alpha)$, $b = \Phi^{-1}(1-\beta)$; Φ^{-1} — Laplacian inverse function. We show that the bound above is achieved in the case of Maximum Distance Separable (MDS) codes $\mathcal{C}$. On the base of this result, we obtain a sufficient number of vectors corresponding to the channel $q\mathrm{SC}(p(2-qp))$. We also present some algorithms for finding the position m_s, the length m_l, and the message $\mathbf{m}$. The main component of them is the algorithm CodeRecognition.

Keywords: *code noising, q-ary symmetric channel, code recognition.*

Введение

Рассмотрим информационно-аналитическую систему (ИАС), в которой два легальных участника (отправитель и получатель) связаны каналом без помех, а пассивный наблюдатель подслушивает передаваемые данные по q-ичному (q — степень простого числа) симметричному каналу с вероятностью p для каждого ненулевого значения ошибки. Эта модель является частным случаем модели канала с наблюдением, впервые рассмотренной в [1]. Как и в [1], предполагается, что отправитель для защиты от наблюдения использует метод кодирования смежными классами (метод кодового зашумления). У наблюдателя при подслушивании одного кодового слова из-за наличия помех возникает неопределённость относительно сообщения, которое было закодировано. Вычислению этой неопределённости в неасимптотическом случае для двоичного симметричного канала наблюдения (при $q = 2$) посвящена работа [2], а в [3] показано, что эта неопределённость может быть снята в рамках модели многократной передачи данных. В частности, в [3] найдена оценка количества зашумлённых кодовых слов, соответствующих одному информационному блоку, достаточного для нахождения этого блока с заданными вероятностями ошибок первого и второго рода. Отметим, что на метод определения информационного блока, основанный на результатах работы [3], накладывается ограничение: наблюдаемые векторы должны соответствовать одному и тому же информационному блоку. В настоящей работе рассматривается более общая задача нахождения информационного блока в рамках модели многократной передачи данных. Предполагается, что различные кодируемые информационные блоки содержат фиксированное сообщение на фиксированной позиции, а наблюдатель получает зашумлённые кодовые слова через q-ичный симметричный канал. В работе ставится задача нахождения позиции и длины подвектора. Зная эти параметры, содержание подвектора может быть определено, например, с помощью метода, основанного на результатах работы [3]. Отметим, что для случая $q = 2$ задача в такой постановке частично решена [4]. Рассмотрение q-ичного симметричного канала представляет не только теоретический интерес. Эта модель канала, например, в последнее время исследуется как адекватная модель пакетной передачи данных [5], а также как модель генерации ошибок в протоколах гомоморфного шифрования [6].

Работа имеет следующую структуру: п. 1 посвящён детальному описанию информационно-аналитической системы. В п. 2 описан метод, разработанный для решения поставленной задачи. Этот метод описан в общем виде, так как может быть применён

для решения других задач, например задач типа распознавания кода. В п. 3 получены оценки применимости разработанного метода в рамках поставленной задачи и построены соответствующие алгоритмы.

1. Информационно-аналитическая система

Пусть информационными блоками являются векторы длины k над полем Галуа $\mathbb{F}_q$ мощности q, где q — степень простого числа. Предполагается, что в момент времени $t \in \mathbb{N}$ информационный блок $\mathbf{s}^{(t)} \in \mathbb{F}_q^k$ имеет вид $\mathbf{s}^{(t)} = (\mathbf{m}_1^{(t)} \parallel \mathbf{m} \parallel \mathbf{m}_2^{(t)})$, где подвектор $\mathbf{m} \in \mathbb{F}_q^{m_l}$ постоянный для всех t, а подвекторы $\mathbf{m}_1^{(t)}$ и $\mathbf{m}_2^{(t)}$ распределены случайно и равновероятно со значениями из $\mathbb{F}_q^{m_s-1}$ и $\mathbb{F}_q^{k-[m_l+m_s]+1}$ соответственно; запись $(\mathbf{a} \parallel \mathbf{b})$ здесь и далее обозначает конкатенацию векторов $\mathbf{a}$ и $\mathbf{b}$. Перед передачей в канал информационные сообщения кодируются отправителем с помощью метода кодового зашумления [2], а именно: для зафиксированных натуральных чисел k и n $(k < n)$ легальными участниками выбирается $((n-k)\times k)$-матрица P с элементами из поля $\mathbb{F}_q$, а каждое сообщение $\mathbf{s}^{(t)}$ длины k кодируется по следующему правилу:

$$\mathrm{Enc}(\mathbf{s}^{(t)}) = (\mathbf{s}^{(t)} + \mathbf{K}^{(t)}P \parallel \mathbf{K}^{(t)}) = \mathbf{c}^{(t)} = (\mathbf{c}_1^{(t)} \parallel \mathbf{K}^{(t)}), \tag{1}$$

где $\mathbf{K}^{(t)}$ — случайно и равновероятно выбранный вектор из $\mathbb{F}_q^{n-k}$. Матрица P и правило кодирования (1) известны всем участникам ИАС (в том числе и наблюдателю), поэтому при отсутствии помех в канале между отправителем и получателем правило декодирования имеет вид $\mathrm{Dec}(\mathbf{c}^{(t)}) = \mathbf{K}^{(t)}P + \mathbf{c}_1^{(t)} = \mathbf{s}^{(t)}$.

Предположим, что наблюдатель передаваемые кодовые слова подслушивает через q-ичный симметричный канал, в котором вероятность возникновения ошибки в каждом символе равна $(q-1)p$, $0 < p < 1/q$, а именно: если $\mathbf{c}^{(t)}$ — вектор длины n на входе канала наблюдения, а $\mathbf{z}^{(t)} = \mathbf{c}^{(t)} + \mathbf{e}^{(t)}$ — вектор на выходе этого канала, то значения координат вектора ошибок $\mathbf{e}^{(t)} = (e_1^{(t)}, \ldots, e_n^{(t)})$ принимают значения в соответствии со следующим распределением вероятностей:

$$\Pr\{\mathbf{e}_i^{(t)} = l\} = \begin{cases} p, & l \in \mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}, \\ 1-(q-1)p, & l = 0, \end{cases} \qquad i \in \{1, \ldots, n\}.$$

В дальнейшем такой q-ичный симметричный канал будем обозначать $q\mathrm{SC}(p)$. Наблюдателю позиции m_s подвектора $\mathbf{m}$ и его длина m_l неизвестны. Целью наблюдателя является нахождение неизвестного подвектора $\mathbf{m}$ при многократной передаче сообщений $\mathbf{s}^{(t)}$, закодированных по правилу (1). Естественно полагать, что эта задача решается наблюдателем последовательно: сначала находятся неизвестные позиция m_s и длина m_l, а затем само сообщение $\mathbf{m}$. Отметим, что при известных m_s и m_l задача нахождения содержания вектора $\mathbf{m}$ может быть решена методом, основанным на результатах работы [3]. Поэтому ниже строится способ нахождения неизвестных значений m_s и m_l.

Наблюдателем выдвигается гипотеза $H_{i,j}$ о том, что $m_s = i$ и $m_l \geqslant j$ (т. е. предполагаемая длина j не превышает истинного значения длины сообщения). Далее для обозначения векторов и матриц в рамках данной гипотезы будем использовать верхний индекс (i,j). В этом случае матрица P представима в блочном виде: $P = [P_1^{(i,j)}|P_2^{(i,j)}|P_3^{(i,j)}]$, где $P_1^{(i,j)}$ — первые $i-1$ столбцов матрицы P; $P_2^{(i,j)}$ — столбцы матрицы P с номерами от i до $i+j-1$; $P_3^{(i,j)}$ — последние $k-(i+j)+1$ столбцов матрицы P. В рамках этой гипотезы наблюдаемый в момент времени t вектор $\mathbf{z}^{(t)} = \mathbf{c}^{(t)} + \mathbf{e}^{(t)}$

имеет следующий вид:

$$\mathbf{z}^{(t)} = (\widehat{\mathbf{m}}_1^{(t)} + \mathbf{K}^{(t)} P_1^{(i,j)} + \mathbf{e}_1^{(t)} \parallel \widehat{\mathbf{m}} + \mathbf{K}^{(t)} P_2^{(i,j)} + \mathbf{e}_2^{(t)} \parallel \widehat{\mathbf{m}}_2^{(t)} + \mathbf{K}^{(t)} P_3^{(i,j)} + \mathbf{e}_3^{(t)} \parallel \mathbf{K}^{(t)} + \mathbf{e}_4^{(t)}), \quad (2)$$

где $\mathbf{e}^{(t)} = (\mathbf{e}_1^{(t)} \parallel \mathbf{e}_2^{(t)} \parallel \mathbf{e}_3^{(t)} \parallel \mathbf{e}_4^{(t)})$ — вектор помехи в канале; $\mathbf{e}_1^{(t)} \in \mathbb{F}_q^{i-1}$; $\mathbf{e}_2^{(t)} \in \mathbb{F}_q^{j}$; $\mathbf{e}_3^{(t)} \in \mathbb{F}_q^{k-(i+j)+1}$; $\mathbf{e}_4^{(t)} \in \mathbb{F}_q^{n-k}$; символ «$\widehat{\ }$» означает, что соответствующие подвекторы могут отличаться от истинных, если гипотеза неверна.

Для подмножества $S = \{i_1, \ldots, i_u\}$ множества $\{1, \ldots, n\}$ рассмотрим линейный оператор проекции $\pi_S : \mathbb{F}^n \to \mathbb{F}^u$, ставящий в соответствие каждому вектору $\mathbf{x} = (x_1, \ldots, x_n)$ из $\mathbb{F}^n$ вектор $\mathbf{x}_S = (x_{i_1}, \ldots, x_{i_u})$. Рассмотрим множество координат $\tau(i,j) = \{i, \ldots, i+j-1\} \cup \{k+1, \ldots, n\}$ и выборку из N векторов (N чётное)

$$\mathbf{Z}_{\tau(i,j)} = \left(\mathbf{z}_{\tau(i,j)}^{(1)}, \mathbf{z}_{\tau(i,j)}^{(2)}, \ldots, \mathbf{z}_{\tau(i,j)}^{(N)}\right), \quad (3)$$

где $\mathbf{z}_{\tau(i,j)}^{(t)} = \pi_{\tau(i,j)}(\widehat{\mathbf{z}}^{(t)})$ — проекция наблюдаемого вектора $\mathbf{z}^{(t)}$ на множество координат $\tau(i,j)$. По выборке (3) построим набор векторов мощности $N/2$:

$$\widetilde{\mathbf{Z}} = (\widetilde{\mathbf{z}}^{(1)}, \widetilde{\mathbf{z}}^{(2)}, \ldots, \widetilde{\mathbf{z}}^{(N/2)}). \quad (4)$$

Здесь $\widetilde{\mathbf{z}}^{(t)} = \mathbf{z}_{\tau(i,j)}^{(2t)} - \mathbf{z}_{\tau(i,j)}^{(2t-1)}$, $t = 1, \ldots, N/2$. С учётом (2), вектор $\widetilde{\mathbf{z}}^{(t)}$ в выборке (4) представим в виде

$$\begin{aligned} \widetilde{\mathbf{z}}^{(t)} &= \left((\mathbf{K}^{(2t)} - \mathbf{K}^{(2t-1)}) P_2^{(i,j)} + \mathbf{e}_2^{(2t)} - \mathbf{e}_2^{(2t-1)} \parallel \mathbf{K}^{(2t)} - \mathbf{K}^{(2t-1)} + \mathbf{e}_4^{(2t)} - \mathbf{e}_4^{(2t-1)}\right) = \\ &= \left(\mathbf{K}^{(2t)} - \mathbf{K}^{(2t-1)}\right) [P_2^{(i,j)} \parallel I_{n-k}] + \left(\mathbf{e}_2^{(2t)} - \mathbf{e}_2^{(2t-1)} \parallel \mathbf{e}_4^{(2t)} - \mathbf{e}_4^{(2t-1)}\right), \end{aligned} \quad (5)$$

где I_{n-k} — единичная матрица. Далее линейный код $\mathcal{C}$ размерности k и длины n будем называть, как обычно, $[n,k]$-кодом [7], а его порождающую матрицу обозначим $G(\mathcal{C})$. Фиксированный набор базисных векторов кода $\mathcal{C}$ обозначим $B_{\mathcal{C}}$, а множество всех различных базисов кода $\mathcal{C}$ — символом $\mathcal{B}_{\mathcal{C}}$.

Лемма 1. Пусть выборка $\mathbf{Z}_{\tau(i,j)}$ вида (3) построена по векторам вида (2), наблюдаемым на выходе $q\mathrm{SC}(p)$; $\mathcal{C}^{(i,j)}$ — $[n-k+j, n-k]$-код с порождающей матрицей $G(\mathcal{C}^{(i,j)}) = [P_2^{(i,j)} | I_{n-k}]$; $H_{i,j}$ — выдвигаемая гипотеза.

1) Если $i = m_s$ и $j \leqslant m_l$, то набор $\widetilde{\mathbf{Z}}$ вида (4) представляет собой набор из зашумлённых кодовых слов кода $\mathcal{C}^{(i,j)}$, полученных из $q\mathrm{SC}(\widehat{p})$, где $\widehat{p} = p(2-pq)$.
2) Пусть множество координат $\{i, \ldots, i+j-1\}$ не вложено в $\{m_s, \ldots, m_s + m_l - 1\}$. Тогда вероятность того, что выборка $\widetilde{\mathbf{Z}}$ представляет собой набор из зашумлённых кодовых слов кода $\mathcal{C}^{(i,j)}$, полученных из $q\mathrm{SC}(\widehat{p})$, не превосходит $(1/q)^{N/2}$.

Доказательство. Так как, по предположению, гипотеза верная, то из (5) следует, что набор (4) является набором из зашумлённых кодовых слов линейного кода $\mathcal{C}^{(i,j)}$ с порождающей матрицей $G(\mathcal{C}^{(i,j)}) = [P_2^{(i,j)} | I_{n-k}]$.

Покажем, что $\Pr\{e_i = l\} = \widehat{p}$ для всех $i \in \{1, \ldots, n\}$, $l \in \mathbb{F}_q^*$. Поскольку в (5) участвует разность помех, реализованных в канале $q\mathrm{SC}(p)$, то по формуле полной вероятности вероятность $\Pr\{e_i = l\}$ для ненулевого l равна сумме вероятностей всевозможных разностей помех, которые в результате дают l. Таких разностей ровно q, причём разностей, содержащих нулевое значение, ровно две: $l-0$ и $0-(q-l)$. Остальные разности

содержат ненулевые значения в качестве уменьшаемого и вычитаемого, которые равновероятны (с вероятностью p), поэтому каждая пара реализуется с вероятностью p^2, их количество равно $q-2$. Таким образом,

$$\widehat{p} = 2(1-(q-1)p)p + \sum_{m=1}^{q-2} p^2 = 2p - 2(q-1)p^2 + p^2(q-2) = p(2-pq).$$

Докажем второе утверждение леммы. Пусть R — множество координат из $\{i, \ldots, i+j-1\}$, которые отсутствуют в $\{m_s, \ldots, m_s + m_l - 1\}$. Тогда вектор из выборки (4) представляет собой сумму зашумлённого кодового слова кода $\mathcal{C}^{(i,j)}$ и случайного равновероятного вектора $\mathbf{g}$ веса $|R|$ (координаты этого вектора представляют собой координаты разности векторов $\mathbf{m}_1^{(2t)} - \mathbf{m}_1^{(2t-1)}$ или $\mathbf{m}_2^{(2t)} - \mathbf{m}_2^{(2t-1)}$). Таким образом, для того чтобы вектор из выборки (4) представлял собой зашумлённое кодовое слово кода $\mathcal{C}^{(i,j)}$, вектор $\mathbf{g}$ должен быть нулевым. Вероятность этого события для одного вектора равна $(1/q)^{|R|}$. Тогда для всей выборки вероятность соответствующего события есть $(1/q)^{|R|N/2}$. Наибольшее значение этой вероятности достигается при $|R| = 1$. ■

Отметим, что функция $g(x) = x(2-xq)$ при каждом натуральном $q \geqslant 2$ возрастает от 0 до $1/q$ на отрезке $[0, 1/q]$. Следовательно, набор (4) представляет собой набор кодовых слов, более зашумлённых, чем слова из набора (3).

В силу леммы 1, проверка верности гипотез наблюдателем сводится к задаче распознавания кода по зашумлённому набору векторов. Заметим, что эта задача имеет несколько способов решения [8, 9]. В настоящей работе применяется метод, схожий с методом из работы [8], идея которого состоит в том, что вес синдрома зашумлённого в двоичном симметричном канале ($q = 2$) кодового слова в среднем меньше веса синдрома произвольного (случайно выбранного) вектора. Обобщённый метод для любого допустимого q приведён далее.

2. Метод распознавания кода в канале $q\mathrm{SC}(p)$

Рассмотрим линейный $[n, k, d]$-код $\mathcal{C}$ ($[n, k]$-код с кодовым расстоянием d) и ему дуальный $[n, n-k]$-код $\mathcal{C}^\perp$. Запись $(\mathbf{a}, \mathbf{b})$ обозначает скалярное произведение векторов $\mathbf{a}$ и $\mathbf{b}$, а для произвольного вектора $\mathbf{h} \in \mathbb{F}_q^n$ символ $\mathbf{h}^\perp$ обозначает подпространство размерности $n-1$, ортогональное вектору $\mathbf{h}$.

Лемма 2. Пусть $\mathbf{h} \in \mathbb{F}_q^n$, $\mathbf{c} \in \mathbf{h}^\perp$, $\mathbf{z} = \mathbf{c} + \mathbf{e}$ — вектор на выходе $q\mathrm{SC}(p)$. Тогда

$$\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = q^{-1}(1 + (q-1)(1-pq)^{\mathrm{w}(\mathbf{h})}), \tag{6}$$

где $\mathrm{w}(\mathbf{h})$ — вес Хэмминга вектора $\mathbf{h}$.

Доказательство. Справедлива следующая цепочка равенств:

$$\begin{gathered}\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = \Pr\{(\mathbf{c}+\mathbf{e}, \mathbf{h}) = 0\} = \Pr\{(\mathbf{e}, \mathbf{h}) = 0\} = \\ = \Pr\{\mathbf{e} \in \mathbf{h}^\perp\} = \sum_{i=0}^{n} A_i^\perp p^i (1-(q-1)p)^{n-i} = A_{\mathbf{h}^\perp}(p, 1-(q-1)p),\end{gathered} \tag{7}$$

где $A_{\mathbf{h}^\perp}(x, y) = \sum\limits_{i=0}^{n} A_i^\perp x^i y^{n-i}$ — энумератор весов кода $\mathbf{h}^\perp$; $A_i^\perp$ — число векторов веса i в коде $\mathbf{h}^\perp$. Для $[n, 1]$-кода $\langle \mathbf{h} \rangle$, порождённого вектором $\mathbf{h}$, рассмотрим энумератор весов $A_{\mathbf{h}}(x, y)$. По определению $A_{\mathbf{h}}(x, y) = \sum\limits_{i=0}^{n} A_i x^i y^{n-i}$, где A_i — число векторов веса i

в коде $\langle \mathbf{h} \rangle$. Так как $\dim(\langle \mathbf{h} \rangle) = 1$, то $A_{\mathbf{h}}(x, y) = y^n + (q-1)x^{\mathrm{w}(\mathbf{h})}y^{n-\mathrm{w}(\mathbf{h})}$. Воспользуемся тождеством Мак-Вильямс и получим

$$A_{\mathbf{h}^\perp}(x, y) = \frac{1}{q}((y + (q-1)x)^n + (q-1)(y-x)^{\mathrm{w}(\mathbf{h})}(y + (q-1)x)^{n-\mathrm{w}(\mathbf{h})}.$$

С учётом (7), подставив $x = p$ и $y = 1 - (q-1)p$ в $A_{\mathbf{h}^\perp}(x, y)$, получим (6). ■

Лемма 3. Пусть $\mathbf{z}$ — случайно и равновероятно выбранный из $\mathbb{F}_q^n$ вектор. Тогда $\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = q^{-1}$.

Доказательство. $\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = \Pr\{\mathbf{z} \in \mathbf{h}^\perp\}$. Но $\Pr\{\mathbf{z} \in \mathbf{h}^\perp\} = \dfrac{|\mathbf{h}^\perp|}{|\mathbb{F}_q^n|} = q^{-1}$. ■

Лемма 4. Пусть $\mathbf{c}$ — вектор, случайно и равновероятно выбранный из кода C' длины n; $C' \not\subseteq C$; $\mathbf{z} = \mathbf{c} + \mathbf{e}$ — вектор на выходе канала $q\mathrm{BC}(p)$. Тогда в любом базисе кода $C^\perp$ найдётся такой $\mathbf{h}$, что $\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = q^{-1}$.

Доказательство. Так как $C' \not\subseteq C$, в любом базисе кода $C^\perp$ всегда найдётся такой вектор $\mathbf{h}$, что $\mathbf{h} \notin C'^\perp$. Выберем такой $\mathbf{h}$. Пусть $\mathbf{h}_{\mathbf{a}}^\perp$ — это смежный класс подпространства $\mathbf{h}^\perp$ вида $\mathbf{h}_{\mathbf{a}}^\perp = \mathbf{a} + \mathbf{h}^\perp$, $\mathbf{a} \in \mathbb{F}_q^n$. Так как $\dim(\mathbf{h}^\perp) = n - 1$, то $|\mathbb{F}_q^n / \mathbf{h}^\perp| = q$. Рассмотрим множество $\mathcal{R} \subset \mathbb{F}_q^n$ мощности $q-1$, такое, что $\left(\bigcup\limits_{\mathbf{a}\in\mathcal{R}} \mathbf{h}_{\mathbf{a}}^\perp\right) \cup \mathbf{h}^\perp = \mathbb{F}_q^n$. Символом $-\mathbf{a}$ обозначим такой вектор, что $-\mathbf{a} + \mathbf{a} = \mathbf{0} \in \mathbb{F}_q^n$. Так как $-\mathbf{a} \in \mathcal{R}$ для любого $\mathbf{a} \in \mathcal{R}$, то

$$\begin{gathered}
\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = \Pr\{\mathbf{z} \in \mathbf{h}^\perp\} = \Pr\{\mathbf{c} + \mathbf{e} \in \mathbf{h}^\perp\} = \\
= \Pr\{\mathbf{c} \in \mathbf{h}^\perp\} \Pr\{\mathbf{e} \in \mathbf{h}^\perp\} + \sum_{\mathbf{a}\in\mathcal{R}} \Pr\{\mathbf{c} \in \mathbf{h}_{\mathbf{a}}^\perp\} \Pr\{\mathbf{e} \in \mathbf{h}_{-\mathbf{a}}^\perp\} = \\
= \Pr\{\mathbf{c} \in C' \cap \mathbf{h}^\perp\} \Pr\{\mathbf{e} \in \mathbf{h}^\perp\} + \sum_{\mathbf{a}\in\mathcal{R}} \Pr\{\mathbf{c} \in C' \cap \mathbf{h}_{\mathbf{a}}^\perp\} \Pr\{\mathbf{e} \in \mathbf{h}_{-\mathbf{a}}^\perp\} = \\
= \frac{|\mathbf{h}^\perp \cap C'|}{|C'|} \Pr\{\mathbf{e} \in \mathbf{h}^\perp\} + \sum_{\mathbf{a}\in\mathcal{R}} \frac{|\mathbf{h}_{\mathbf{a}}^\perp \cap C'|}{|C'|} \Pr\{\mathbf{e} \in \mathbf{h}_{-\mathbf{a}}^\perp\}.
\end{gathered}$$

Учитывая равенство $|\mathbf{h}_{\mathbf{a}}^\perp \cap C'| = |\mathbf{h}^\perp \cap C'|$ для всех $\mathbf{a} \in \mathcal{R}$, получим

$$\Pr\{(\mathbf{z}, \mathbf{h}) = 0\} = \frac{|\mathbf{h}^\perp \cap C'|}{|C'|} \left(\Pr\{\mathbf{e} \in \mathbf{h}^\perp\} + \sum_{\mathbf{a}\in\mathcal{R}} \Pr\{\mathbf{e} \in \mathbf{h}_{-\mathbf{a}}^\perp\}\right) = \frac{|\mathbf{h}^\perp \cap C'|}{|C'|}.$$

Пусть размерность кода C' равна r. Так как $\dim(\mathbf{h}^\perp) = n-1$ и $C' \not\subseteq \mathbf{h}^\perp$ (по построению $\mathbf{h} \notin C'^\perp$), то $\dim(\mathbf{h}^\perp \cap C') = r - 1$. Тогда $\dfrac{|\mathbf{h}^\perp \cap C'|}{|C'|} = \dfrac{q^{r-1}}{q^r} = \dfrac{1}{q}$. Отсюда следует доказываемое утверждение. ■

Пусть $\mathbf{h} \in \mathbb{F}_q^n$, $\mathbf{z} \in \mathbb{F}_q^n$. Рассмотрим случайную величину $X_{\mathbf{h},\mathbf{z}}$, принимающую значение 0, если $(\mathbf{h}, \mathbf{z}) = 0$, и значение 1 при $(\mathbf{h}, \mathbf{z}) \neq 0$. Если $\mathbf{c} \in \mathbf{h}^\perp$, $\mathbf{z} = \mathbf{c} + \mathbf{e}$ — вектор на выходе $q\mathrm{BC}(p)$, то соответствующую случайную величину $X_{\mathbf{h},\mathbf{z}}$ будем обозначать $X_{\mathbf{h},\mathbf{z}}^0$; если $\mathbf{c}' \in C' \not\subseteq \mathbf{h}^\perp$, $\mathbf{z} = \mathbf{c}' + \mathbf{e}$ — вектор на выходе $q\mathrm{BC}(p)$ или если $\mathbf{z}$ — случайно и равновероятно выбранный из $\mathbb{F}_q^n$ вектор, то случайную величину обозначим $X_{\mathbf{h},\mathbf{z}}^1$. Пусть $\mathbf{p}_0 = \Pr\{X_{\mathbf{h},\mathbf{z}}^0 = 1\}$, тогда из (6) получаем

$$\mathbf{p}_0 = 1 - \frac{1}{q}\left(1 + (q-1)(1-pq)^{\mathrm{w}(\mathbf{h})}\right) = \frac{q-1}{q}(1 - (1-pq)^{\mathrm{w}(\mathbf{h})}). \tag{8}$$

Пусть $\mathbf{p}_1 = \Pr\{X^1_{\mathbf{h},\mathbf{z}} = 1\}$. Из лемм 3 и 4 получаем

$$\mathbf{p}_1 = 1 - \frac{1}{q} = \frac{q-1}{q}. \tag{9}$$

Для $\mathbf{h} \in \mathbb{F}_q^n$, $\mathcal{S} \subseteq \mathbb{F}_q^n$ и $T \geqslant 0$ обозначим через $\mathrm{ST}(\mathbf{h}, \mathcal{S}, T)$ статистический критерий, согласно которому принимается решение о том, что выборка $\mathcal{S}$ является набором зашумлённых векторов из $\mathbf{h}^\perp$, если $\sum\limits_{\mathbf{z}\in\mathcal{S}} X_{\mathbf{h},\mathbf{z}} \leqslant T$. Рассмотрим случайные величины $X = \sum\limits_{\mathbf{z}\in\mathcal{S}}^{M} X_{\mathbf{h},\mathbf{z}}$ и $X_i = \sum\limits_{\mathbf{z}\in\mathcal{S}}^{M} X^i_{\mathbf{h},\mathbf{z}}$, $i = 0, 1$.

Теорема 1. Пусть $\mathbf{h} \in \mathbb{F}_q^n$, $\mathcal{S}$ — набор $M_{\mathbf{h}}$ векторов, полученный из канала $q\mathrm{SC}(p)$. При выборе статистического критерия $\mathrm{ST}(\mathbf{h}, \mathcal{S}, T_{\mathbf{h}})$ проверки гипотезы $H_0 : X = X_0$ против альтернативы $H_1 : X = X_1$ вероятности ошибок первого и второго рода не превышают α и β соответственно для

$$M_{\mathbf{h}} = \left(\frac{b\sqrt{(q-1)(1-(1-pq)^{\mathrm{w}(\mathbf{h})})(1+(q-1)(1-pq)^{\mathrm{w}(\mathbf{h})})} - a}{(\sqrt{q-1})(1-pq)^{\mathrm{w}(\mathbf{h})}} \right)^2, \tag{10}$$

$$T_{\mathbf{h}} = M_{\mathbf{h}} \frac{q-1}{q} + a\sqrt{M_{\mathbf{h}} \frac{q-1}{q^2}},$$

где $a = \Phi^{-1}(\alpha)$; $b = \Phi^{-1}(1-\beta)$; Φ^{-1} — обратная функция Лапласа.

Доказательство. Для каждого $i \in \{0, 1\}$ случайная величина X_i является суммой бернуллиевских случайных величин. Тогда по центральной предельной теореме случайная величина X_i асимптотически нормальна и её математическое ожидание равно $M_{\mathbf{h}}\mathbf{p}_i$, а дисперсия — $M_{\mathbf{h}}\mathbf{p}_i(1-\mathbf{p}_i)$. Пусть α и β — выбранные ошибки первого и второго рода соответственно при использовании статистического критерия $\mathrm{ST}(\mathbf{h}, \mathcal{S}, T_{\mathbf{h}})$. Тогда при проверке гипотезы $H_0 : X = X_0$ против альтернативы $H_1 : X = X_1$ имеют место равенства

$$\Pr\left\{\sum_{\mathbf{z}\in\mathcal{S}} X^1_{\mathbf{h},\mathbf{z}} \leqslant T_{\mathbf{h}}\right\} = \alpha, \quad \Pr\left\{\sum_{\mathbf{z}\in\mathcal{S}} X^0_{\mathbf{h},\mathbf{z}} \leqslant T_{\mathbf{h}}\right\} = 1-\beta.$$

Учитывая асимптотическую нормальность случайных величин X_0 и X_1, получим

$$\begin{cases} \Phi\left(\dfrac{T_{\mathbf{h}} - M_{\mathbf{h}}\mathbf{p}_1}{\sqrt{M_{\mathbf{h}}\mathbf{p}_1(1-\mathbf{p}_1)}}\right) = \alpha, \\ \Phi\left(\dfrac{T_{\mathbf{h}} - M_{\mathbf{h}}\mathbf{p}_0}{\sqrt{M_{\mathbf{h}}\mathbf{p}_0(1-\mathbf{p}_0)}}\right) = 1-\beta, \end{cases}$$

или

$$\begin{cases} T_{\mathbf{h}} = M_{\mathbf{h}}\mathbf{p}_1 + \Phi^{-1}(\alpha)\sqrt{M_{\mathbf{h}}\mathbf{p}_1(1-\mathbf{p}_1)}, \\ T_{\mathbf{h}} = M_{\mathbf{h}}\mathbf{p}_0 + \Phi^{-1}(1-\beta)\sqrt{M_{\mathbf{h}}\mathbf{p}_0(1-\mathbf{p}_0)}. \end{cases}$$

В результате имеем

$$\begin{cases} T_{\mathbf{h}} = M_{\mathbf{h}}\mathbf{p}_1 + \Phi^{-1}(\alpha)\sqrt{M_{\mathbf{h}}\mathbf{p}_1(1-\mathbf{p}_1)}, \\ M_{\mathbf{h}} = \left(\dfrac{\Phi^{-1}(1-\beta)\sqrt{\mathbf{p}_0(1-\mathbf{p}_0)} - \Phi^{-1}(\alpha)\sqrt{\mathbf{p}_1(1-\mathbf{p}_1)}}{\mathbf{p}_1 - \mathbf{p}_0}\right)^2. \end{cases}$$

Подставив значение $\mathbf{p}_0$ и $\mathbf{p}_1$ из (8) и (9), получим искомые выражения для $T_{\mathbf{h}}$ и $M_{\mathbf{h}}$. ■

Замечание 1. В доказательстве теоремы используется аппроксимация функции распределения F случайной величины X_i, $i \in \{0,1\}$, к функции Лапласа Ф. Поэтому в полученных формулах возникают погрешности, влияющие на доверительные вероятности. В соответствии с неравенством Берри — Эссеена [10, с. 75] погрешность для α меньше, чем $\dfrac{\mathbf{p}_0^2 + (1-\mathbf{p}_0)^2}{\sqrt{\mathbf{p}_0(1-\mathbf{p}_0)M_{\mathbf{h}}}}$, а для β меньше, чем $\dfrac{\mathbf{p}_1^2 + (1-\mathbf{p}_1)^2}{\sqrt{\mathbf{p}_1(1-\mathbf{p}_1)M_{\mathbf{h}}}}$.

Для каждого $p \in (0, 1/q)$ и $q \geqslant 2$ рассмотрим непрерывную функцию

$$A(x) = 1 + (q-2)(1-pq)^x - (q-1)(1-pq)^{2x}.$$

По построению, функция $A(x)$ для натуральных x всегда положительная. Непосредственно подставив в (10) выражение для $\mathbf{p}_1$ из формулы (9) и проведя необходимые преобразования, получим

$$M_{\mathbf{h}} = \left(\frac{b\sqrt{A(\mathrm{w}(\mathbf{h}))} - a}{\sqrt{q-1}(1-pq)^{\mathrm{w}(\mathbf{h})}} \right)^2.$$

Рассмотрим функцию $f(x) = \dfrac{bA(x)^{1/2} - a}{\sqrt{q-1}(1-pq)^x}$. Производная $f'(x)$ имеет вид

$$f'(x) = \frac{b(A(x)^{1/2})'\sqrt{q-1}(1-pq)^x - (bA(x)^{1/2} - a)\sqrt{q-1}(1-pq)^x \ln(1-pq)}{(q-1)(1-pq)^{2x}} =$$
$$= \frac{b(A(x)^{1/2})' - (bA(x)^{1/2} - a)\ln(1-pq)}{\sqrt{q-1}(1-pq)^x} =$$
$$= \frac{(-b/2A(x)^{1/2} - b/2A(x)^{-1/2} - b/2A(x)^{-1/2}(q-1)(1-pq)^{2x} + a)\ln(1-pq)}{\sqrt{q-1}(1-pq)^x}.$$

Отсюда получаем, что при $p \in (0, 1/q)$ и $q \geqslant 2$ производная равна нулю ($f'(x) = 0$) тогда и только тогда, когда

$$A(x)^{1/2} + A(x)^{-1/2} + A(x)^{-1/2}(q-1)(1-pq)^{2x} = \frac{2a}{b}. \tag{11}$$

Заметим, что $A(x)^{1/2} + A(x)^{-1/2}(1 + (q-1)(1-pq)^{2x}) \geqslant A(x)^{1/2} + A(x)^{-1/2} \geqslant 2$, так как функция $A(x)$ положительная для натуральных x и $y + 1/y \geqslant 2$ для $y > 0$. Если $a/b < 1$, то уравнение (11) решений не имеет, поэтому функция $f(x)$ (а также и $f^2(x)$) либо монотонно убывает, либо монотонно возрастает для $x > 0$. Для произвольных $p \in (0, 1/q)$ и $q \geqslant 2$ вычисления показывают, что $f(1) \leqslant f(2)$, следовательно, функция $f(x)$ возрастающая. Таким образом, справедлива

Лемма 5. Пусть $\mathbf{h} \neq \mathbf{h}'$, $w(\mathbf{h}) > w(\mathbf{h}')$ и в рамках условий теоремы 1 выполняется неравенство $a/b < 1$. Тогда $M_{\mathbf{h}} \geqslant M_{\mathbf{h}'}$.

Таким образом, чем меньше вес вектора $\mathbf{h}$, тем меньше может быть мощность выборки $\mathcal{S}$ зашумлённых векторов для применения статистического критерия $\mathrm{ST}(\mathbf{h}, \mathcal{S}, T_{\mathbf{h}})$.

Ниже приведён алгоритм 1 проверки гипотезы H_0 против альтернативы H_1. Естественно считать, что чем меньший размер выборки требуется для применения статистического критерия при заданных уровнях ошибок первого и второго рода, тем эффективнее критерий. В соответствии с леммой 5, для применения статистического критерия следует выбирать базисные векторы малого веса. Для этого нам понадобится

понятие минимального базиса кода, а именно: минимальным базисом кода $\mathcal{C}$ будем называть такой базис $B^0_{\mathcal{C}} \in \mathcal{B}_{\mathcal{C}}$, что в любом другом базисе $B_{\mathcal{C}} \in \mathcal{B}_{\mathcal{C}}$ найдётся базисный вектор, вес которого не меньше веса любого вектора из $B^0_{\mathcal{C}}$. Отметим, что минимальный базис определяется неоднозначно.

Алгоритм 1. CodeRecognition$(\mathcal{C}, \mathcal{S}, p, \alpha, \beta)$ (распознавание кода)

Вход: $\mathcal{C}$ — предполагаемый код, $\mathcal{S}$ — выборка, p — вероятность ошибки в $q\mathrm{SC}(p)$, α и β — доверительные вероятности ошибок 1 и 2 рода для (10).

1: Найти минимальный базис $B^0_{\mathcal{C}^\perp}$ кода $\mathcal{C}^\perp$.
2: **Для всех** $\mathbf{h} \in B^0_{\mathcal{C}^\perp}$
3: найти $M_{\mathbf{h}}$ и $T_{\mathbf{h}}$ согласно теореме 1;
4: выбрать набор $\mathcal{S}_{\mathbf{h}}$ из $\mathcal{S}$ мощности $M_{\mathbf{h}}$;
5: проверить с помощью критерия $\mathrm{ST}(\mathbf{h}, \mathcal{S}_{\mathbf{h}}, T_{\mathbf{h}})$ принадлежность выборки $\mathcal{S}_{\mathbf{h}}$ каждому пространству $\mathbf{h}^\perp$.
6: **Если** выборка $\mathcal{S}_{\mathbf{h}}$ является набором зашумлённых векторов из $\mathbf{h}^\perp$ для каждого $\mathbf{h}$ из $B^0_{\mathcal{C}^\perp}$, **то**
7: выборка $\mathcal{S}$ полагается набором зашумлённых кодовых слов кода $\mathcal{C}$, в противном случае считается, что $\mathcal{S}$ не является набором зашумлённых кодовых слов кода $\mathcal{C}$.

На шаге 1 алгоритма предлагается искать минимальный базис, так как в этом случае потребуется наименьшее в рамках предлагаемого способа количество наблюдаемых векторов. Для произвольного кода задача нахождения минимального базиса представляется трудной, но для некоторых кодов такой базис может быть найден легко. К последним относятся коды с максимальным достижимым расстоянием (МДР-коды) и равновесные коды. Так как порождающая матрица линейного $[n, n-k]$-кода $\mathcal{C}^\perp$ может быть приведена к псевдосистематическому виду [7], то вес каждого вектора в $B^0_{\mathcal{C}^\perp}$ не превышает $k+1$. Из лемм 2 и 5 следует

Теорема 2. Пусть $\mathcal{C}$ — $[n, k]$-код, $M(\mathcal{C}, \alpha, \beta)$ — минимально необходимое количество векторов, достаточное для проверки алгоритмом CodeRecognition гипотезы H_0 против гипотезы H_1. Тогда $M(\mathcal{C}, \alpha, \beta) \leqslant f^2(k+1)$, причём равенство выполняется в случае, когда $\mathcal{C}$ — МДР-код.

3. Поиск подвектора в подслушанных данных

Рассмотрим ИАС, описанную в п. 1. Предлагается следующий способ (алгоритм 2) определения длины и позиции сообщения $\mathbf{m}$, основанный на лемме 1. Он заключается в том, что если выборка $\widetilde{\mathbf{Z}}$ вида (4) для гипотезы $H_{i,j}$ представляет собой набор кодовых слов кода $\mathcal{C}^{(i,j)}$, то гипотеза $H_{i,j}$ верна. Тот факт, что выборка (4) является набором зашумлённых кодовых слов кода $\mathcal{C}^{(i,j)}$, можно проверить с помощью статистического критерия $\mathrm{ST}(\mathbf{h}, \widetilde{\mathbf{Z}}, T)$, где $\mathbf{h}$ пробегает базис кода $(\mathcal{C}^{(i,j)})^\perp$, а объём выборки $\widetilde{\mathbf{Z}}$ и параметр T для каждого $\mathbf{h}$ определяются в соответствии с теоремой 1. Таким образом, проверяя гипотезы, можно найти длину m_l и позицию m_s сообщения $\mathbf{m}$. Заметим, что при проверке текущей гипотезы $H_{i,j}$ необязательно перехватывать новые сообщения, так как можно использовать уже перехваченные сообщения, использовавшиеся для проверки предыдущих гипотез. Если же для какой-либо гипотезы недостаточно перехваченных ранее сообщений, то необходимо дополнительно перехватить ровно столько, сколько не хватило.

Поясним алгоритм ShiftAndLenFinding. Сначала ищется начальная координата неизвестного вектора $\mathbf{m}$, при этом на основании п. 1 леммы 1 полагаем, что длина этого

Алгоритм 2. ShiftAndLenFinding($\mathcal{S}, p, \alpha, \beta, P$) (нахождение длины и позиции)

Вход: $\mathcal{S}$ — выборка, p — вероятность ошибки в qSC(p), α и β — доверительные вероятности ошибок 1 и 2 рода для (10), P — матрица из (1).

1: $i := 1$, $j := 1$, $m_s := -1$.
2: В рамках гипотезы $H_{i,j}$ построить матрицу $[P_2^{(i,j)}|I_{n-k}]$ и найти $\widehat{p}$ (см. лемму 1).
3: Для кода $\mathcal{C}^{(i,j)}$ с порождающей матрицей $G(\mathcal{C}^{(i,j)}) = [P_2^{(i,j)}|I_{n-k}]$ вычислить $M = \max\{M_{\mathbf{h}} : \mathbf{h} \in B^0_{(\mathcal{C}^{(i,j)})^\perp}\}$ в соответствии с (10) для $p = \widehat{p}$.
4: **Если** $|\mathcal{S}| < 2M$, **то** перехватить недостающее количество векторов.
5: Построить выборку $\mathcal{S}^{(i,j)}$ вида (4).
6: Проверить гипотезу с помощью алгоритма CodeRecognition($\mathcal{C}^{(i,j)}, \mathcal{S}^{(i,j)}, \widehat{p}, \alpha, \beta$).
7: **Если** $m_s > 0$, **то** перейти на шаг 9.
8: **Если** гипотеза неверна, **то** $i := i + 1$ и возврат к шагу 2, **иначе** $m_s := i$.
9: **Если** гипотеза верна, **то** $j := j + 1$ и возврат к шагу 2, **иначе** $m_l := j - 1$.

Выход: m_s, m_l — позиция и длина сообщения $\mathbf{m}$.

вектора равна 1. На этапе нахождения начальной координаты гипотеза H_0 проверяется против альтернативы H_1, так как при ошибочном выборе начальной координаты (и при предполагаемой единичной длине секретного подвектора) набор (4) представляет собой набор случайных и равновероятных векторов (при условии, что ключи $\mathbf{K}^{(t)}$ и векторы $\mathbf{m}_i^{(t)}$, $i \in \{1, 2\}$, случайны и равновероятны, см. п. 1). Когда предполагаемая позиция вектора $\mathbf{m}$ найдена, начинается этап поиска длины путём постепенного увеличения, начиная с единицы. При этом если длина будет превышать истинную длину вектора $\mathbf{m}$, то набор (4) уже будет представлять собой зашумлённые векторы некоторого кода, содержащего код $\mathcal{C}^{(i,j)}$. В частности, если $i = m_s$ и $j = m_l + 1$ (то есть позиция вектора угадана верно, но предполагаемая длина на единицу больше, чем истинная), построенная выборка (4) будет представлять собой векторы линейного $[n-k+j, n-k]$-кода, порождённого базисом кода $\mathcal{C}^{(m_s, m_l+1)}$ и вектором $(\mathbf{0}^1 \parallel 1 \parallel \mathbf{0}^2)$, где $\mathbf{0}^1$ и $\mathbf{0}^2$ — нулевые векторы длины m_l и $n - k$ соответственно. Но так как вектор $\mathbf{m}_2^{(t)}$ случайный и равновероятный, каждый вектор в наборе (4) с вероятностью $1 - 1/q^2$ принадлежит смежному классу $(\mathbf{0}^1 \parallel 1 \parallel \mathbf{0}^2) + \mathcal{C}^{(m_s, m_l+1)}$ кода $\mathcal{C}^{(m_s, m_l+1)}$. При правильно найденных длине m_l и позиции m_s выборка $\mathbf{Z}_{\tau(m_s, m_l)}$ вида (3) представляет собой набор векторов вида

$$\widehat{\mathbf{z}}^{(t)} = (\mathbf{m} + \mathbf{K}^{(t)} P_2^{(m_s, m_l)} \parallel \mathbf{K}^{(t)}) + (\mathbf{e}_2^{(t)} \parallel \mathbf{e}_4^{(t)}), \quad t = 1, 2, \ldots \tag{12}$$

Тогда задача восстановления сообщения $\mathbf{m}$ может быть решена, например, полным перебором по всевозможным значениям из $\mathbb{F}_q^{m_l}$ с помощью алгоритма CodeRecognition (если размер поля q и длина m_l сообщения позволяют произвести полный перебор за приемлемое время). Соответствующий алгоритм 3 построен ниже.

Так как $\widehat{p} \geqslant p$ (п. 1 леммы 1), для применения алгоритма ShiftAndLenFinding потребуется как минимум в 2 раза больше перехватов, чем для применения алгоритма SubvectorFinding. Таким образом, при нахождении содержания сообщения может быть использована выборка, построенная во время выполнения ShiftAndLenFinding. Для матрицы P (см. правило (1)) обозначим $M(P)$ минимальное число векторов, достаточное для нахождения алгоритмами ShiftAndLenFinding и SubvectorFinding вектора $\mathbf{m}$.

Алгоритм 3. SubvectorFinding($\mathcal{S}, p, \alpha, \beta$) (нахождение значения вектора)

Вход: $\mathcal{S} = \{\mathbf{z}^{(t)}\}_{t=1,2,\ldots}$ — выборка вида (3) при $i = m_s$ и $j = m_l$, p — вероятность ошибки в qSC(p), α и β — вероятности ошибок 1 и 2 рода для (10).

1: $\mathcal{M} := \varnothing$.
2: **Для всех** $\mathbf{y} \in \mathbb{F}_q^{m_l}$
3: по выборке $\mathcal{S}$ построить выборку $\widehat{\mathcal{S}} = \{\widehat{\mathbf{z}}^{(t)}\}_{t=1,2,\ldots}$, где $\widehat{\mathbf{z}}^{(t)} = \mathbf{z}^{(t)} - (\mathbf{y} \parallel \mathbf{0})$, $\mathbf{0}$ — нулевой вектор пространства $\mathbb{F}_q^{n-k}$;
4: проверить с помощью алгоритма CodeRecognition($\mathcal{C}^{(m_s,m_l)}, \widehat{\mathcal{S}}, p, \alpha, \beta$) гипотезу о том, что выборка $\widehat{\mathcal{S}}$ порождена кодовыми словами кода $\mathcal{C}^{(m_s,m_l)}$.
5: **Если** гипотеза верна, **то** $\mathcal{M} := \mathcal{M} \cup \{\mathbf{y}\}$.

Выход: $\mathcal{M}$.

Замечание 2. Если $\mathbf{m} \neq \mathbf{y}$, но $(\mathbf{m} - \mathbf{y} \parallel \mathbf{0}) \in \mathcal{C}^{(m_s,m_l)}$, то алгоритм SubvectorFinding добавит вектор $\mathbf{y}$ в множество кандидатов $\mathcal{M}$. Так как порождающая матрица $[n - k + m_l, n - k]$-кода $\mathcal{C}^{(m_s,m_l)}$ имеет вид $G(\mathcal{C}^{(m_s,m_l)}) = [P_2^{(m_s,m_l)}|I_{n-k}]$, то $(\mathbf{m} - \mathbf{y} \parallel \mathbf{0}) \in \mathcal{C}^{(m_s,m_l)}$ только в случае $\mathbf{m} = \mathbf{y}$. Поэтому существуют такие значения α и β, при которых алгоритм SubvectorFinding возвращает одноэлементное множество кандидатов.

Отметим, что нахождение сообщения $\mathbf{m}$ по выборке $\mathbf{Z}_{\tau(m_s,m_l)}$, состоящей из векторов вида (12), возможно и неперебо́рным методом. Один из таких методов может быть построен для двоичного случая ($q = 2$) на основе результатов работы [3].

Для примера рассмотрим $[n, n-k]$-код $\mathcal{C}$ Рида — Соломона над полем $\mathbb{F}_q$, $q > 2$. Пусть порождающая матрица кода $\mathcal{C}$ имеет вид $G(\mathcal{C}) = [P|I_{n-k}]$. Заметим, что код $\mathcal{C}$ — это МДР-код, поэтому для любого $\omega \subset \{1, \ldots, k\}$ и $\tau = \omega \cup \{k+1, \ldots, n\}$ код $\pi_\tau(\mathcal{C})$ с порождающей матрицей $\pi_\tau(G(\mathcal{C})) = [\pi_\omega(P)|I_{n-k}]$ является МДР-кодом размерности $n-k$ и длины $|\omega|+n-k$. Так как $\pi_\tau(G(\mathcal{C}))^\perp$ — также МДР-код, его кодовое расстояние равно $n-k+1$ (не зависит от ω и его мощности). Пусть в правиле (1) используется матрица P, являющаяся частью порождающей матрицы $G(\mathcal{C})$ рассматриваемого кода Рида — Соломона $\mathcal{C}$. Тогда построенная на шаге 2 алгоритма ShiftAndLenFinding матрица $\pi_{\tau(i,j)}(P) = [P_2^{(i,j)}|I_{n-k}]$ является матрицей МДР-кода $\pi_{\tau(i,j)}(\mathcal{C})$ размерности $n-k+1$ и по теореме 2 минимальный базис $B^0_{\pi_{\tau(i,j)}(\mathcal{C})^\perp}$, строящийся на первом шаге алгоритма CodeRecognition (к этому алгоритму обращение происходит на шаге 6 алгоритма определения позиции и длины постоянного подвектора), состоит из векторов веса $n-k+1$. Отметим, что базис $B^0_{\pi_{\tau(i,j)}(\mathcal{C})^\perp}$, например, может представлять собой строки матрицы $[I_j|(-P_2^{(i,j)})^\top]$ — порождающей матрицы кода $\pi_{\tau(i,j)}(\mathcal{C})^\perp$, где $A^\top$ — транспонированная матрица A. Таким образом, размер выборки для определения позиции и длины сообщения, а также содержания сообщения определяется на основании длины ключа, используемого в правиле (1). Результаты вычислений минимального количества перехваченных векторов для нахождения подвектора $\mathbf{m}$ в зашумленных данных приведены в таблице для $q = 16$, $n = 15$, $\alpha = 0{,}05$, $\beta = 0{,}04$.

Значение величны $M(P)$ в случае, когда $[P|I_{n-k}]$ — порождающая матрица кода Рида — Соломона, $q = 16$, $n = 15$, $\alpha = 0{,}05$, $\beta = 0{,}04$

	p							
$n-k$	$\leqslant 0{,}0125$	0,01875	0,025	0,03125	0,0375	0,04375	0,05	0,05625
1	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	213
3	2	2	2	2	2	5	519	2,0 E+6
4	2	2	2	2	6	494	3,2 E+5	2,1 E+10
5	2	2	2	3	194	60604	2,0 E+8	2,1 E+14
6	2	2	2	37	7515	7,5 E+6	1,3 E+11	2,1 E+18
7	2	2	4	570	293173	9,2 E+8	7,9 E+13	2,1 E+22
8	2	2	28	9083	1,1 E+7	1,1 E+11	4,9 E+16	2,1 E+26
9	2	2	207	145186	4,5 E+8	1,4 E+13	3,1 E+19	2,1 E+30
10	2	4	1585	2,3 E+6	1,7 E+10	1,7 E+15	1,9 E+22	2,1 E+34
11	2	15	12205	3,7 E+7	6,8 E+11	2,1 E+17	1,2 E+25	2,1 E+38
12	2	59	94107	5,9 E+8	2,7 E+13	2,6 E+19	7,5 E+27	2,1 E+42
13	2	242	725943	9,5 E+9	1,0 E+15	3,3 E+21	4,7 E+30	2,1 E+46
14	2	1003	5,6 E+6	1,5 E+11	4,1 E+16	4,0 E+23	2,9 E+33	2,1 E+50

Теорема 3. Пусть q — степень простого числа; $q\mathrm{SC}(p)$ — q-ичный симметричный канал с вероятностью ошибки p; $n, k \in \mathbb{N}$, $k < n$; $\alpha, \beta \in [0; 1]$; P и $\widetilde{P}$ — разные $((n-k) \times k)$-матрицы, причём матрица $[P|I_{n-k}]$ является порождающей матрицей МДР-кода. Тогда $M(P) \geqslant M(\widetilde{P})$.

Доказательство. Ранее показано, что для МДР-кода минимальный базис дуального кода состоит из векторов веса $n-k+1$. Заметим, что для не-МДР-кода минимальный базис дуального кода состоит из векторов веса, не превышающего $n-k+1$. На основании теоремы 2 получаем доказываемое утверждение. ■

Таким образом, при заданных q, p, n, k, α и β лучшую стойкость метода кодового зашумления к атаке нахождения фиксированного вектора обеспечивают такие $((n-k) \times k)$-матрицы P, для которых матрица $[P|I_{n-k}]$ порождает МДР-код, так как в этом случае потребуется наибольшее количество перехватов для нахождения сообщения.

ЛИТЕРАТУРА

1. *Wyner A. D.* The wire-tap channel // Bell Sys. Tech. J. 1975. V. 54. P. 1355–1387.
2. *Коржик В. И., Яковлев В. А.* Неасимптотические оценки эффективности кодового зашумления одного канала // Пробл. передачи информ. 1981. Т. 17. № 4 С. 11–18.
3. *Иванов В. А.* Статистические методы оценки эффективности кодового зашумления // Труды по дискретной математике. Т. 6. М.: Физматлит, 2002. С. 48–63.
4. *Косолапов Ю. В., Турченко О. Ю.* Поиск информационного сообщения в зашумлённых кодовых блоках при многократной передаче данных // Прикладная дискретная математика. Приложение. 2016. № 9. С. 55–57.
5. *Weidmann C.* Coding for the q-ary symmetric channel with moderate q // IEEE Int. Symp. Inform. Theory. 2008. P. 2156–2159.
6. *Couvreur A.* Distinguisher-based attacks on public-key cryptosystems using Reed — Solomon codes // Designs, Codes and Cryptography. 2014. V. 73. No. 2. P. 641–666.
7. *Сидельников В. М.* Теория кодирования. М.: Физматлит, 2008. 324 с.
8. *Chabot C.* Recognition of a code in a noisy environment // Proc. IEEE ISIT. June 2007. P. 2211–2215.

9. *Yardi A. D. and Vijayakumaran S.* Detecting linear block codes in noise using the GLRT // Proc. IEEE Intern. Conf. Communications, Budapest, Hungary, June 9–13, 2013. P. 4895–4899.
10. *Ширяев А. Н.* Вероятность. В 2-х кн. 3-е изд., перераб. и доп. М.: МЦНМО, 2004. Кн. 1 — 520 с., кн. 2 — 408 с.

REFERENCES

1. *Wyner A. D.* The wire-tap channel. Bell Sys. Tech. J., 1975, vol. 54, pp. 1355–1387.
2. *Korzhik V. I. and Yakovlev V. A.* Neasimptoticheskie otsenki effektivnosti kodovogo zashumleniya odnogo kanala [Nonasymptotic estimates for efficiency of code jamming in a wire-tap channel]. Probl. Peredachi Inf., 1981, vol. 17, iss. 4, pp. 11–18. (in Russian)
3. *Ivanov V. A.* Statisticheskie metody otsenki effektivnosti kodovogo zashumleniya [Statistical estimation of the code noising efficiency]. Tr. Diskr. Mat., 2002, vol. 6, pp. 48–63. (in Russian)
4. *Kosolapov Y. V. and Turchenko O. Y.* Poisk informatsionnogo soobshcheniya v zashumlennykh kodovykh blokakh pri mnogokratnoy peredache dannykh [Search of an information message in noisy code blocks at repeated data transmission]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2016, no. 9, pp. 55–57. (in Russian)
5. *Weidmann C.* Coding for the q-ary symmetric channel with moderate q. IEEE Int. Symp. Inform. Theory, 2008, pp. 2156–2159.
6. *Couvreur A.* Distinguisher-based attacks on public-key cryptosystems using Reed — Solomon codes. Designs, Codes and Cryptography, 2014, vol. 73, no. 2, pp. 641–666.
7. *Sidel'nikov V. M.* Teoriya Kodirovaniya [Coding Theory]. Moscow, Fizmatlit Publ., 2008, 324 p. (in Russian)
8. *Chabot C.* Recognition of a code in a noisy environment. Proc. IEEE ISIT, June 2007, pp. 2211–2215.
9. *Yardi A. D. and Vijayakumaran S.* Detecting linear block codes in noise using the GLRT. Proc. IEEE Intern. Conf. Communications, Budapest, Hungary, June 9–13, 2013, pp. 4895–4899.
10. *Shiryaev A. N.* Veroyatnost' [Probability]. Moscow, MCCME Publ., 2004. (in Russian)

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.1

УСЛОВИЯ ПРИМИТИВНОСТИ И ОЦЕНКИ ЭКСПОНЕНТОВ МНОЖЕСТВ ОРИЕНТИРОВАННЫХ ГРАФОВ

Я. Э. Авезова*, В. М. Фомичев*,**,***

*Национальный исследовательский ядерный университет «МИФИ», г. Москва, Россия
**Финансовый университет при Правительстве Российской Федерации, г. Москва, Россия
***Институт проблем информатики ФИЦ ИУ РАН, г. Москва, Россия

Исследованы вопросы минимизации заданного примитивного множества неотрицательных матриц порядка n (n-вершинных орграфов), где минимизация понимается как определение минимальных примитивных подмножеств. Получен универсальный критерий примитивности множества орграфов $\hat{\Gamma} = \{\Gamma_1, \ldots, \Gamma_p\}$, $p > 1$, выраженный через характеристики мультиграфа $\Gamma_1 \cup \ldots \cup \Gamma_p$, в котором каждая дуга орграфа Γ_i помечена символом i, $i = 1, \ldots, p$. Показано, что задача распознавания примитивности множества орграфов алгоритмически разрешима. Для частного класса множеств, когда орграфы $\Gamma_1, \ldots, \Gamma_p$ содержат общее множество контуров, получен ряд достаточных условий примитивности множества $\hat{\Gamma}$. Для множества орграфов $\hat{\Gamma} = \{\Gamma_0, \ldots, \Gamma_{n-1}\}$, где Γ_i — граф с множеством вершин $\{0, \ldots, n-1\}$, имеющий гамильтонов контур $(0, \ldots, n-1)$ и дугу $(i, (i+l) \bmod n)$, где $n \geqslant l > 1$, $i = 0, \ldots, n-1$, уточнён критерий примитивности (множество орграфов $\hat{\Gamma}$ примитивное тогда и только тогда, когда НОД$(n, l-1) = 1$) и в случае примитивности получены оценки экспонента: $n-1 \leqslant \exp \hat{\Gamma} \leqslant 2n-2$. Минимальное примитивное подмножество множества $\hat{\Gamma}$, на котором достигается верхняя оценка экспонента, содержит не более n/d орграфов, где $d =$ НОД(n, l).

Ключевые слова: *граф Виландта, примитивное множество матриц (графов), экспонент графа.*

DOI 10.17223/20710410/35/8

CONDITIONS OF PRIMITIVITY AND EXPONENT BOUNDS FOR SETS OF DIGRAPHS

Y. E. Avezova*, V. M. Fomichev*,**,***

*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russia
**Financial University under the Government of the Russian Federation, Moscow, Russia
***The Institute of Informatics Problems of the Russian Academy of Sciences, Moscow, Russia

E-mail: avezovayana@gmail.com, fomichev.2016@yandex.ru

For a set of digraphs $\hat{\Gamma} = \{\Gamma_1, \ldots, \Gamma_p\}$, $p > 1$, we present a criterion to be primitive. We do it in terms of characteristics of the multidigraph $\Gamma^{(p)} = \Gamma_1 \cup \ldots \cup \Gamma_p$ where each edge in Γ_i is assigned the label i, $i = 1, \ldots, p$. Any walk of length s in $\Gamma^{(p)}$ is assigned a word $w = w_1 \ldots w_s$ of length s over the alphabet $\{1, \ldots, p\}$, and the corresponding

product of digraphs $\Gamma(w) = \Gamma_{w_1} \cdot \ldots \cdot \Gamma_{w_s}$ is introduced. The walk is assigned the label w^t if it is the concatenation of t walks labeled with w. The multidigraph $\Gamma^{(p)}$ is called w-strongly connected if it is strongly connected and, for all its vertices i and j, there exists a walk in $\Gamma^{(p)}$ from i to j labeled with w^t for some natural number t. By the definition, the set of digraphs $\hat{\Gamma}$ is primitive if and only if $\Gamma(w)$ is primitive for some word w. Thus, we have the following criterion: the digraph $\Gamma(w)$ is primitive if and only if $\Gamma^{(p)}$ is w-strongly connected and has cycles labeled with $w^{t_1}, \ldots, w^{t_m}$, where $\gcd(t_1, \ldots, t_m) = 1$. As a corollary, we prove that the problem of recognizing the primitivity of $\hat{\Gamma}$ is algorithmically decidable. In the particular case, when the digraphs in $\hat{\Gamma}$ have the common set of cycles $\hat{C} = \{C_1, \ldots, C_m\}$ of lengths $l_1, \ldots, l_m$ respectively, $m \geqslant 1$, $l_1 \leqslant \ldots \leqslant l_m$, the digraph $\Gamma(w)$, $w = w_1 \ldots w_s$, is primitive if any one of the following conditions holds: a) $m = 1$ and $l_1 = 1$; b) $\gcd(l_1, \ldots, l_m) = s$; c) the digraph $C_1 \cup \ldots \cup C_m$ is connected and has quasi-Eulerian $\hat{C}$-cycle of length s. At last, for the set of digraphs $\hat{\Gamma} = \{\Gamma_0, \ldots, \Gamma_{n-1}\}$ with vertex set $\{0, \ldots, n-1\}$, where for some l, $n \geqslant l > 1$, each Γ_i, $i \in \{0, \ldots, n-1\}$, has a Hamiltonian cycle $(0, \ldots, n-1)$ and the edge $(i, (i + l) \mod n)$, we prove the following criterion of primitivity and bounds for the exponent: the set $\hat{\Gamma} = \{\Gamma_0, \ldots, \Gamma_{n-1}\}$ is primitive if and only if $\gcd(n, l-1) = 1$, and in this case $n - 1 \leqslant \exp\hat{\Gamma} \leqslant 2n - 2$. The minimal subset of $\hat{\Gamma} = \{\Gamma_0, \ldots, \Gamma_{n-1}\}$ with exponent $2n - 2$ contains at most n/d digraphs, where $d = \gcd(n, l)$. The presented results are used for evaluating mixing properties of cryptographic functions compositions.

Keywords: *Wielandt's graph, primitive set of matrices (digraphs), exponent of digraph.*

Введение

Введём основные обозначения:

- $\mathbb{N}$ — множество натуральных чисел;
- $N_p = \{1, \ldots, p\}$, $p \in \mathbb{N}$;
- $V = \{1, \ldots, n\}$ — множество вершин орграфа;
- X^* — множество всех слов в конечном алфавите X;
- НОД$(a_1, \ldots, a_n)$ — наибольший общий делитель натуральных чисел $a_1, \ldots, a_n$;
- (i, j) — дуга в орграфе Γ, инцидентная вершинам i и j;
- $[i, j]$ — путь в орграфе Γ из вершины i в вершину j;
- $\langle Y \rangle$ — полугруппа, порождённая подмножеством Y мультипликативной полугруппы;
- $M_0(n)$ — множество всех $0, 1$-матриц порядка n;
- $\Gamma(n)$ — множество всех n-вершинных орграфов.

Понятие примитивности (множественной) для множества ориентированных графов и неотрицательных матриц впервые введено В. Н. Сачковым в [1]. Множество матриц $\hat{M}$ называется множественно примитивным, если при некотором $l \in \mathbb{N}$ положительны все слова длины l в алфавите $\hat{M}$. Наименьшее такое l называется множественным экспонентом множества $\hat{M}$. Множественный экспонент любого множественно примитивного множества не превышает $2^n - 2$. В [2] даны оценки множественного экспонента для некоторых классов множеств орграфов.

Исследуемое в работе свойство примитивности множества матриц введено в [3] и определено как существование положительного слова в алфавите $\hat{M}$, где слово называется положительным, если произведение составляющих его матриц есть положи-

тельная матрица. Экспонентом примитивного множества $\hat{M}$ называется наименьшая длина положительного слова в алфавите $\hat{M}$.

Данная работа посвящена вопросам минимизации заданного примитивного множества $\hat{M}$ неотрицательных матриц порядка n (n-вершинных орграфов), где минимизация понимается как определение минимальных примитивных подмножеств множества $\hat{M}$. В [4] введено понятие сокращённого множества матриц и доказано, что экспонент множества матриц равен экспоненту его сокращённого подмножества. Таким образом, исследование примитивности множества матриц можно свести к исследованию примитивности сокращённых множеств. Заметим, что при сокращении исходного множества матриц (орграфов) величина экспонента может увеличиться. Понятие сокращённого множества обобщено, что позволяет ограничить исследование примитивности множества матриц ещё более узким классом множеств.

Построен критерий примитивности множества орграфов $\hat{\Gamma} = \{\Gamma_1, \ldots, \Gamma_p\}$, $p > 1$, в терминах свойств мультиграфа $\Gamma_1 \cup \ldots \cup \Gamma_p$. Показано, что задача распознавания примитивности множества орграфов алгоритмически разрешима. Для частного класса множеств, когда орграфы $\Gamma_1, \ldots, \Gamma_p$ содержат общее множество контуров, получен ряд достаточных условий примитивности множества $\hat{\Gamma}$.

В силу широкого применения в криптосистемах регистровых преобразований особый интерес представляет исследование примитивности множества графов, соответствующих регистрам сдвига. Для частного класса таких множеств уточнён критерий примитивности, получена оценка экспонента.

1. Минимизация примитивного множества матриц

Обозначим: $\hat{\Gamma} = \{\Gamma_1, \ldots, \Gamma_p\}$ — множество орграфов с множеством вершин V, где все дуги орграфа Γ_r помечены числом r; $\hat{M} = \{M_1, \ldots, M_p\}$ — множество 0,1-матриц, где $M_r = (m_{ij}(r))$ — матрица смежности вершин орграфа Γ_r, $r = 1, \ldots, p$, $p > 1$.

Пусть $w = w_1 \ldots w_s$ — слово в алфавите N_p. Тогда слову $(M_{w_1}, \ldots, M_{w_s}) \in \langle \hat{M} \rangle$ (слову $(\Gamma_{w_1}, \ldots, \Gamma_{w_s}) \in \langle \hat{\Gamma} \rangle$) соответствует произведение матриц $M(w) = M_{w_1} \ldots M_{w_s}$ (произведение орграфов $\Gamma(w) = \Gamma_{w_1} \ldots \Gamma_{w_s}$); положим $M(w) = (m_{ij}(w))$. Слово $(M_{w_1}, \ldots, M_{w_s})$ назовём положительным (примитивным), если матрица $M(w)$ положительная (примитивная). Слово $(\Gamma_{w_1}, \ldots, \Gamma_{w_s})$ назовём положительным (примитивным), если орграф $\Gamma(w)$ полный (примитивный). Множество $\hat{M}$ (множество $\hat{\Gamma}$) называется примитивным, если полугруппа $\langle \hat{M} \rangle$ содержит положительное слово (полугруппа $\langle \hat{\Gamma} \rangle$ содержит полный орграф), наименьшая длина положительного слова называется экспонентом множества $\hat{M}$ (множества $\hat{\Gamma}$) (обозначается $\exp \hat{M}$ и $\exp \hat{\Gamma}$ соответственно).

Отметим важные свойства множеств $M_0(n)$ и $\Gamma(n)$ [3, 5]:

1) На множестве $M_0(n)$ задан частичный порядок: $M \leqslant M' \Leftrightarrow m_{ij} \leqslant m'_{ij}$ для всех $i, j = 1, \ldots, n$. Если при этом существуют такие i и j, что $m_{ij} < m'_{ij}$, то пишем $M < M'$. Пусть M, M' — матрицы смежности орграфов Γ, Γ' соответственно и $M \leqslant M'$, тогда все дуги орграфа Γ являются дугами орграфа Γ', т. е. Γ — часть орграфа Γ' (обозначается $\Gamma \leqslant \Gamma'$).
2) На множестве всех подмножеств множества $\Gamma(n)$ задан квазипорядок: $\hat{\Gamma} \geqslant \hat{\Gamma}'$ для данных множеств $\hat{\Gamma}, \hat{\Gamma}' \Leftrightarrow$ для любого орграфа $\Gamma' \in \hat{\Gamma}'$ имеется орграф $\Gamma \in \hat{\Gamma}$, где $\Gamma \geqslant \Gamma'$. Если $\hat{\Gamma} \geqslant \hat{\Gamma}'$, то $\exp \hat{\Gamma} \leqslant \exp \hat{\Gamma}'$.
3) Если множество $\hat{\Gamma}$ примитивное, то орграф $\Gamma = \Gamma_1 \cup \ldots \cup \Gamma_p$ также примитивный и $\exp \Gamma \leqslant \exp \hat{\Gamma} \leqslant \min\{\exp \Gamma_1, \ldots, \exp \Gamma_p\}$.

Пусть $\hat{M}$ — примитивное множество матриц из $M_0(n)$, $\hat{\Gamma}$ — примитивное множество n-вершинных орграфов, $M \in \hat{M}$, $\Gamma \in \hat{\Gamma}$. Обозначим: $\hat{M}\backslash M$ — подмножество $\hat{M}$, полученное удалением из $\hat{M}$ матрицы M; $\hat{\Gamma}\backslash\Gamma$ — подмножество $\hat{\Gamma}$, полученное удалением из $\hat{\Gamma}$ орграфа Γ. Слово $(M_{w_1}, \ldots, M_{w_s})$ в алфавите $\hat{M}\backslash M$ назовём покрывающим для M, если $M \leqslant M(w)$. Слово $(\Gamma_{w_1}, \ldots, \Gamma_{w_s})$ в алфавите $\hat{\Gamma}\backslash\Gamma$ назовём покрывающим для Γ, если $\Gamma \leqslant \Gamma(w)$. Матрицу M (орграф Γ), для которой существует покрывающее слово, назовём избыточной матрицей в множестве $\hat{M}$ (избыточным орграфом в множестве $\hat{\Gamma}$). Множество $\hat{M}$ (множество $\hat{\Gamma}$) называется минимальным, если оно не содержит избыточной матрицы (орграфа). В частности, если множество $\hat{M}$ содержит примитивную матрицу M, то минимальным является одноэлементное множество $\{M\}$.

Задача минимизации примитивного множества $\hat{M}$ (множества $\hat{\Gamma}$) состоит в определении его минимальных примитивных подмножеств.

Утверждение 1. Матрица M — избыточная в примитивном множестве $\hat{M}$, если и только если подмножество $\hat{M}\backslash M$ примитивное.

Доказательство. Необходимость. Пусть M — избыточная в примитивном множестве $\hat{M}$ и $(M_{w_1}, \ldots, M_{w_s})$ — покрывающее слово для матрицы M, $M(w) = M_{w_1} \ldots M_{w_s}$. Так как множество $\hat{M}$ примитивное, то существует положительное слово в алфавите $\hat{M}$. Заменим в этом слове каждое вхождение символа M на слово $(M_{w_1}, \ldots, M_{w_s})$, получим слово в алфавите $\hat{M}\backslash M$. Поскольку $M \leqslant M(w)$, полученное слово положительное. Следовательно, подмножество $\hat{M}\backslash M$ примитивное.

Достаточность. Если $\hat{M}\backslash M$ примитивное, то существует некоторое положительное слово в алфавите $\hat{M}\backslash M$, а следовательно, и в алфавите $\hat{M}$. ■

Матрица M называется максимальной матрицей множества $\hat{M}$, если для матрицы $M' \in \hat{M}$ из соотношения $M \leqslant M'$ следует $M = M'$. Множество матриц $\hat{M}$ называется сокращённым, если оно состоит только из максимальных матриц [4]. Аналогично определяются понятия максимального орграфа и сокращённого множества орграфов. Любое множество матриц (орграфов) может быть приведено к сокращённому удалением всех немаксимальных матриц (орграфов).

Если множество матриц $\hat{M}$ (орграфов $\hat{\Gamma}$) не сокращённое, то в $\hat{M}$ ($\hat{\Gamma}$) есть такие матрицы M, M' (орграфы Γ, Γ'), что $M < M'$ ($\Gamma < \Gamma'$). Тогда M' (Γ') есть покрывающее слово длины 1 для M (Γ). Следовательно, минимальное множество матриц (орграфов) является сокращённым.

Множество $\langle M_0(n), \leqslant\rangle$ с введённым выше отношением частичного порядка образует решётку порядка 2^m, где $m = n^2$, изоморфную решётке двоичных n^2-мерных векторов. Количество положительных элементов матрицы M назовём её весом. Слоем с номером k решетки $\langle M_0(n), \leqslant\rangle$ назовём её подмножество, состоящее из матриц веса k, $k = 0, 1, \ldots, m$.

Утверждение 2. Пусть $\hat{M} = \{M_1, \ldots, M_p\}$ — сокращенное множество матриц. Тогда:

1) $\hat{M}$ образует антицепь в решётке $\langle M_0(n), \leqslant\rangle$ и $p \leqslant C_m^{\lceil m/2 \rceil}$;
2) при $p = 2$ существует $2^{2m-1} + 2^{m-1} - 3^m$ сокращённых множеств матриц.

Доказательство.

1) По определению любые две различные максимальные матрицы несравнимы. Значит, сокращённое множество состоит из попарно несравнимых матриц.

Количество матриц в антицепи не превышает ширину решетки $\langle M_0(n), \leqslant\rangle$, обозначаемую h, где $h = C_m^{\lceil m/2 \rceil}$ [3, с. 25].

2) Пусть M — матрица веса k. Тогда существует 2^{m-k} матриц, которые не меньше (поэлементно) матрицы M, и (2^k-1) матриц, которые поэлементно меньше матрицы M. Следовательно, для каждой матрицы веса k существует $r_k = 2^m - 2^{m-k} - 2^k + 1$ матриц, которые с ней несравнимы. Число матриц в k-м слое решётки равно C_m^k. Тогда общее количество упорядоченных пар несравнимых матриц равно

$$\begin{aligned}\sum_{k=1}^{m-1} C_m^k r_k = 2^m \sum_{k=1}^{m-1} C_m^k - \sum_{k=1}^{m-1} C_m^k 2^{m-k} - \sum_{k=1}^{m-1} C_m^k 2^k + \sum_{k=1}^{m-1} C_m^k = \\ = 2^m(2^m-2) - 2(3^m - 2^m - 1) + (2^m - 2) = 2^{2m} - 2\cdot 3^m + 2^m.\end{aligned} \tag{1}$$

Поскольку матрицы не упорядочены, число сокращённых множеств при $p=2$ вдвое меньше (1), т. е. равно $2^{2m-1} + 2^{m-1} - 3^m$. ■

Определим условия, при которых сокращённое множество из двух матриц $\hat{M} = \{A, B\}$ не является минимальным. Это означает, что матрицы A и B несравнимы, но существует такое натуральное t, что $A^t \geqslant B$ (либо $B^t \geqslant A$), и примитивность множества равносильна примитивности матрицы A (матрицы B).

Для оценки t воспользуемся понятием локальной примитивности матриц и орграфов [6]. Пусть $I = \{i_1, \ldots, i_k\}$, $J = \{j_1, \ldots, j_r\}$, $\varnothing \neq I, J \subseteq N_n$, A — матрица порядка n, $A(I \times J)$ — её подматрица размера $k \times r$, полученная из A удалением строк с номерами $i \notin I$ и столбцов $j \notin J$. Матрица A называется $I \times J$-примитивной, если существует число $\gamma \in \mathbb{N}$, такое, что матрица $A^t(I \times J)$ положительна при любом $t \geqslant \gamma$. Наименьшее такое число γ называется $I \times J$-экспонентом матрицы A, обозначается через $I \times J$-exp A. Орграф Γ называется $I \times J$-примитивным, если при некотором $\gamma \in \mathbb{N}$ для любых $(i,j) \in I \times J$ в Γ имеются пути длины $t \geqslant \gamma$ из i в j, наименьшее такое t равно $I \times J$-exp Γ. Если A — матрица смежности орграфа Γ, то A примитивная тогда и только тогда, когда примитивный Γ, и $I \times J$-exp $A = I \times J$-exp Γ. При $I = \{i\}$, $J = \{j\}$ обозначим $i \times j$-exp $A = I \times J$-exp A.

Далее $\Gamma(A)$ и $\Gamma(B)$ — n-вершинные орграфы с матрицами смежности $A = (a_{ij})$ и $B = (b_{ij})$ соответственно, $A^t = (a_{i,j}^{(t)})$, $t \geqslant 1$. Множество дуг орграфа $\Gamma(B)$ обозначим $E(B)$.

Утверждение 3. Пусть матрицы A и B несравнимы. Отношение $A^t \geqslant B$ выполнено при $t \in \mathbb{N}$, если и только если в $\Gamma(A)$ имеется путь длины t из i в j при любой паре $(i,j) \in E(B)$. В частности, если матрица A является $i \times j$-примитивной при любой паре $(i,j) \in E(B)$, то

$$t \leqslant \max_{(i,j) \in E(B)} i \times j\text{-exp } A.$$

Доказательство. Для выполнения условия $A^t \geqslant B$ необходимо и достаточно, чтобы $a_{i,j}^{(t)} = 1$ при любой паре $(i,j) \in E(B)$. В соответствии с теоремой, устанавливающей связь между орграфами и их матрицами смежности [7, с. 143], в матрице A^t элемент $a_{i,j}^{(t)}$ равен 1 тогда и только тогда, когда в $\Gamma(A)$ есть путь длины t из вершины i в вершину j. Следовательно, $A^t \geqslant B$ тогда и только тогда, когда в $\Gamma(A)$ есть путь длины t из вершины i в вершину j при любой паре $(i,j) \in E(B)$.

Если матрица A является $i \times j$-примитивной при любой паре $(i,j) \in E(B)$, то при любом $\tau \geqslant \max\limits_{(i,j) \in E(B)} i \times j$-exp$A$ выполнено $A^\tau \geqslant B$. Следовательно, $t \leqslant \max\limits_{(i,j) \in E(B)} i \times j$-exp A. ■

Замечание 1. Для выполнения условия $A^t \geqslant B$ при любом $t \leqslant \max\limits_{(i,j) \in E(B)} i \times j$-exp A необходимо, чтобы при любой паре $(i,j) \in E(B)$ в орграфе $\Gamma(A)$ существовал путь

из вершины i в вершину j, проходящий через вершину некоторого сильносвязного подграфа орграфа $\Gamma(A)$ [6, с. 73]. Достаточные условия $i\times j$-примитивности орграфа и оценки $i\times j$-экспонента приведены в [8].

2. Критерий примитивности множества графов

Если множество $\hat{\Gamma}$ примитивное, то мультиграф $\Gamma^{(p)}$ сильносвязный [3, 5]. В $\Gamma^{(p)}$ любой путь длины s помечен некоторым словом w длины s из N_p^*. В частности, путь в мультиграфе $\Gamma^{(p)}$ имеет метку w^t, если он есть конкатенация t путей с метками w; путь с меткой w^0 положим пустым.

Утверждение 4. В мультиграфе $\Gamma^{(p)}$ есть путь $[i,j]$ с меткой w^t, где $w = w_1 \dots w_s$, если и только если в орграфе $\Gamma(w)$ есть путь $[i,j]$ длины t.

Доказательство. В мультиграфе $\Gamma^{(p)}$ число путей длины l из вершины i в вершину j с меткой w равно $m_{ij}(w)$ [3, с. 57]. Отсюда в мультиграфе $\Gamma^{(p)}$ имеется путь $[i,j]$ с меткой w, если и только если в орграфе $\Gamma(w)$ есть дуга (i,j). Следовательно, в мультиграфе $\Gamma^{(p)}$ есть путь $[i,j]$ с меткой w^t, если и только если в орграфе $\Gamma(w^t)$ есть дуга (i,j), т. е. в орграфе $\Gamma(w)$ есть путь $[i,j]$ длины t. ■

Сильносвязный мультиграф $\Gamma^{(p)}$ назовём w-сильносвязным, если для любых $i,j\in V$ существует путь с меткой $w^{t_{ij}}$ из i в j при некотором $t_{ij}\in\mathbb{N}$. Из данного определения следует, что в w-сильносвязном мультиграфе $\Gamma^{(p)}$ есть контур с меткой w^{t_i} при некотором $t_i\in\mathbb{N}$ с началом обхода в вершине i, $i=1,\dots,n$. В частности, при $t_i=1$ мультиграф $\Gamma^{(p)}$ содержит контур с меткой w и, в соответствии с утверждением 4, $\Gamma(w)$ имеет петлю в вершине i.

В мультиграфе $\Gamma^{(p)}$ обозначим $V_w(i)$ — множество вершин, до которых существует путь с меткой w с началом в вершине $i\in V$.

Пример 1. На рис. 1 изображены множество графов $\hat{\Gamma}=\{\Gamma_1,\Gamma_2\}$ и $(1,2)$-сильносвязный мультиграф $\Gamma^{(2)}$, где $V_w(1)=\{3\}$, $V_w(2)=\{4,5\}$, $V_w(3)=\{1,2\}$, $V_w(4)=\{3\}$, $V_w(5)=\{4,5\}$.

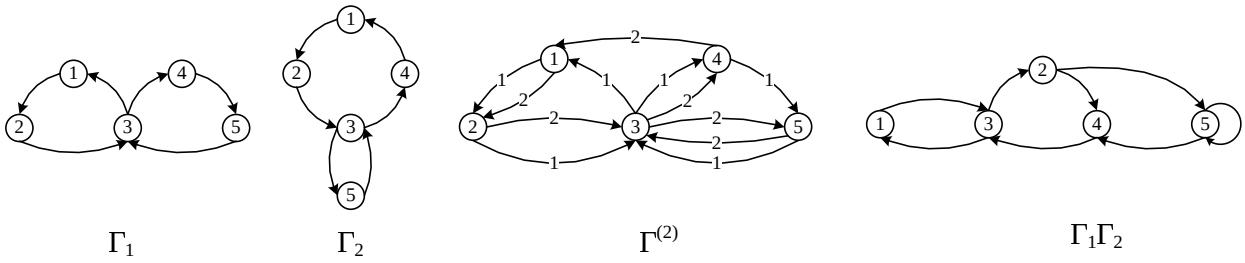

Рис. 1. Иллюстрация к примеру 1

Утверждение 5. В w-сильносвязном мультиграфе $\Gamma^{(p)}$:

а) $V_w(i)\neq\varnothing$ для любой вершины $i\in V$;

б) если $i\in V_w(i)$, то $|V_w(i)|>1$;

в) $\bigcup\limits_{i\in V} V_w(i)=V$.

Доказательство. В w-сильносвязном мультиграфе $\Gamma^{(p)}$ для любых вершин $i,j\in V$ существует путь с меткой $w^{t_{ij}}$ из вершины i в вершину j при некотором $t_{ij}\in\mathbb{N}$. Этот путь есть конкатенация t_{ij} путей с меткой w, первый из которых имеет начальную вершину i, а последний — конечную вершину j. Поскольку вершина i (вершина j) является начальной (конечной) вершиной пути с меткой $w^{t_{ij}}$, то она является

начальной (конечной) и для первого (последнего) пути в рассматриваемой конкатенации путей с меткой w. Следовательно, в w-сильносвязном орграфе $\Gamma^{(p)}$ любая вершина является начальной для некоторого пути с меткой w, а также любая вершина является конечной для некоторого пути с меткой w.

Если $V_w(i) = \varnothing$ для некоторой вершины $i \in V$, то в $\Gamma^{(p)}$ не существует пути с меткой w с началом в вершине i; если $V_w(i) = \{i\}$, то для любой вершины $j \neq i$ в $\Gamma^{(p)}$ не существует пути с меткой w с началом в вершине i; если $\bigcup\limits_{i \in V} V_w(i) = Q$ и $Q \subset V$, то вершины множества $V \backslash Q$ не являются конечными ни для одного из путей с меткой w; в каждом из трёх случаев имеем противоречие с w-сильной связностью мультиграфа $\Gamma^{(p)}$. ■

Обозначим $C(i)$ — контур (не обязательно простой) с началом обхода из вершины i. Выполнено необходимое условие w-сильной связности мультиграфа $\Gamma^{(p)}$.

Утверждение 6. Если мультиграф $\Gamma^{(p)}$ w-сильносвязный, то для любой его вершины i существует контур $C(i)$ с меткой w^{t_i}, проходящий через все вершины $\Gamma^{(p)}$.

Доказательство. В w-сильносвязном мультиграфе $\Gamma^{(p)}$ существуют пути $[1,2]$ с меткой $w^{t_{12}}$ и $[2,1]$ с меткой $w^{t_{21}}$, пути $[1,3]$ с меткой $w^{t_{13}}$ и $[3,1]$ с меткой $w^{t_{31}}$, …, пути $[1,n]$ с меткой $w^{t_{1n}}$ и $[n,1]$ с меткой $w^{t_{n1}}$. Конкатенация этих путей в названном порядке образует контур $C(1)$ с меткой w^{t_1}, где $t_1 = \sum\limits_{j=2}^{n} t_{1j} + t_{j1}$, проходящий через все вершины $\Gamma^{(p)}$. Аналогичным образом для вершин $i = 2, \ldots, n$ доказывается наличие в $\Gamma^{(p)}$ контура $C(i)$ с меткой w^{t_i}, где $t_i = \sum\limits_{j \in V \backslash \{i\}} t_{ij} + t_{ji}$, проходящего через все вершины $\Gamma^{(p)}$. ■

Замечание 2. Утверждение, обратное к утверждению 6, в общем случае неверно. Проиллюстрируем это на простейшем примере. Рассмотрим двухвершинный мультиграф $\Gamma^{(2)}$ с четырьмя дугами: $(1,2)$ с меткой 1, $(1,2)$ с меткой 2, $(2,1)$ с меткой 1 и $(2,1)$ с меткой 2. Пусть $w = (1,2)$. Тогда в мультиграфе есть контуры $C(1)$, $C(2)$ с меткой w, однако нет путей $[1,2]$ и $[2,1]$ с метками $w^{t_{12}}$ и $w^{t_{21}}$ соответственно ни при каких t_{12}, t_{21}. Следовательно, $\Gamma^{(2)}$ не является w-сильносвязным при $w = (1,2)$.

Справедлив следующий критерий примитивности орграфа $\Gamma(w)$.

Теорема 1. Орграф $\Gamma(w) = \Gamma_{w_1} \ldots \Gamma_{w_s}$, где $w = w_1 \ldots w_s$, примитивный, если и только если $\Gamma^{(p)}$ является w-сильносвязным и содержит контуры с метками $w^{t_1}, \ldots, w^{t_m}$, где $\text{НОД}(t_1, \ldots, t_m) = 1$.

Доказательство. Необходимость. Пусть орграф $\Gamma(w)$ примитивный и $\exp \Gamma(w) = t$, тогда любые две вершины в $\Gamma(w)$ соединены путём длины t и в $\Gamma(w)$ есть множество контуров длин $t_1, \ldots, t_m$, где $\text{НОД}(t_1, \ldots, t_m) = 1$, в частности петля. Из наличия в $\Gamma(w)$ пути длины t между любой парой вершин следует, что в орграфе $\Gamma^{(p)}$ для любых $i, j \in V$ есть путь с меткой w^t из вершины i в вершину j. Следовательно, $\Gamma^{(p)}$ w-сильносвязный.

По утверждению 4, при любом $l \in \mathbb{N}$ в мультиграфе $\Gamma^{(p)}$ есть контур с меткой w^l, если и только если в орграфе $\Gamma(w)$ есть контур длины l. Следовательно, в $\Gamma^{(p)}$ есть контуры с метками $w^{t_1}, \ldots, w^{t_m}$, где $\text{НОД}(t_1, \ldots, t_m) = 1$. В частности, если в $\Gamma(w)$ есть петля, то в $\Gamma^{(p)}$ есть контур с меткой w. В этом случае без ущерба для общности полагаем $t_1 = 1$ и тогда $\text{НОД}(1, t_2, \ldots, t_m) = 1$.

Достаточность. Пусть мультиграф $\Gamma^{(p)}$ w-сильносвязный, тогда в нём есть путь с меткой $w^{t_{ij}}$ из вершины i в вершину j для любых i, j. Отсюда следует, что $m_{ij}(w^{t_{ij}})$=1,

т. е. в орграфе $\Gamma(w^{t_{ij}})$ есть дуга (i,j). Следовательно, в $\Gamma(w)$ есть путь $[i,j]$ длины t_{ij} для любых вершин i,j. Значит, орграф $\Gamma(w)$ сильносвязный.

Пусть в $\Gamma^{(p)}$ имеются $m \geqslant 1$ контуров с метками $w^{t_1},\dots,w^{t_m}$. Тогда, по утверждению 4, в $\Gamma(w)$ есть контуры длины $t_1,\dots,t_m$. При НОД$(t_1,\dots,t_m)=1$ орграф $\Gamma(w)$ примитивный. ■

Следствие 1. Задача распознавания примитивности множества n-вершинных орграфов алгоритмически разрешима.

Доказательство. Покажем, что w-сильную связность мультиграфа $\Gamma^{(p)}$ достаточно проверить для конечного множества слов w.

Слова w и w' в алфавите N_p назовём эквивалентными (обозначается $w \cong w'$), если $\Gamma(w)=\Gamma(w')$. Пусть $|\Gamma(n)|=l$, тогда число классов эквивалентности не превышает l.

Докажем (от противного), что в каждом классе эквивалентности содержится слово длины не более l. Пусть в некотором классе эквивалентности K наименьшая длина содержащихся слов равна t, где $l<t$. Запишем такое слово $w=w_1\dots w_t$ и обозначим $w(\tau)=w_1\dots w_\tau$, $\tau=1,\dots,t$. Тогда при $l<t$ в последовательности $\Gamma(w(\tau))$, $\tau=1,\dots,t$, имеются одинаковые орграфы, то есть $\Gamma(w(\tau))=\Gamma(w(\theta))$, где $1\leqslant\tau<\theta\leqslant t$. Значит, $w(\tau)\cong w(\theta)$. Если $\theta=t$, то $w(\tau)\cong w(t)$, и если $\theta<t$, то $w(\tau)\cdot(w_{\theta+1}\dots w_t)\cong w(\theta)\cdot(w_{\theta+1}\dots w_t)$, где символ $\cdot$ обозначает конкатенацию слов. Так как $w(\theta)\cdot(w_{\theta+1}\dots w_t)=w(t)$, в любом случае получаем, что в K имеется слово длины меньше t, что противоречит условию. Следовательно, w-сильную связность мультиграфа $\Gamma^{(p)}$ достаточно проверить для всех слов w длины не более l, при этом для любого слова w задача равносильна распознаванию примитивности орграфа $\Gamma(w)$. ■

Пусть $\hat{C}=\{C_1,\dots,C_m\}$ — множество контуров длин $l_1,\dots,l_m$ соответственно в орграфе Γ. Если орграф $C_1\cup\dots\cup C_m$ связный, то он содержит контур Z, обходящий однократно каждый контур множества $\hat{C}$ и проходящий через каждую дугу столько раз, сколько контуров множества $\hat{C}$ содержат эту дугу [9, с. 79]. Контур Z называется квазиэйлеровым $\hat{C}$-контуром, его длина равна $l_1+\dots+l_m$.

Теорема 2. Пусть мультиграф $\Gamma^{(p)}$ w-сильносвязный, где $w=w_1\dots w_s$, и орграфы $\Gamma_1,\dots,\Gamma_p$ имеют общее множество контуров $\hat{C}=\{C_1,\dots,C_m\}$ длин $l_1,\dots,l_m$ соответственно, где $m\geqslant 1$, $l_1\leqslant\dots\leqslant l_m$. Тогда орграф $\Gamma_{w_1}\dots\Gamma_{w_s}$ примитивный при любом из условий:

а) $m=1$ и $l_1=1$;

б) НОД$(l_1,\dots,l_m)=s$;

в) орграф $C_1\cup\dots\cup C_m$ связный и содержит квазиэйлеров $\hat{C}$-контур длины s.

Доказательство. В соответствии с теоремой 1, достаточно доказать, что в $\Gamma^{(p)}$ имеются контуры с метками $w^{t_1},\dots,w^{t_m}$, где НОД$(t_1,\dots,t_m)=1$, в частности, при $m=1$ — контур с меткой w.

а) Если $m=1$ и $l_1=1$, то в вершине $u\in V$ имеется петля как в орграфах $\Gamma_1,\dots,\Gamma_p$, так и в $\Gamma^{(p)}$ (с любой из меток $1,\dots,p$). Следовательно, в $\Gamma^{(p)}$ конкатенация s петель в вершине u с метками $w_1,\dots,w_s$ образует контур с меткой w.

б) При НОД$(l_1,\dots,l_m)=s$ длины контуров $C_1,\dots,C_m$ можно представить в виде $l_1=st_1$, $l_2=st_2$, …, $l_m=st_m$, где $t_1,\dots,t_m\in\mathbb{N}$ и НОД$(t_1,\dots,t_m)=1$. Контуры $C_1,\dots,C_m$ есть в каждом из графов $\Gamma_1,\dots,\Gamma_p$, следовательно, в $\Gamma^{(p)}$ есть контуры длин $l_1,\dots,l_m$ с любыми метками, в том числе с метками $w^{t_1},\dots,w^{t_m}$, $w=w_1\dots w_s$.

в) Квазиэйлеров контур $\hat{C}$-контур длины s есть в каждом из графов $\Gamma_1, \ldots, \Gamma_p$, следовательно, в $\Gamma^{(p)}$ есть контур с любой меткой длины s, в том числе с меткой w. Без ущерба для общности положим $t_1 = 1$. Следовательно, НОД$(1, t_2, \ldots, t_m) = 1$. ■

3. Оценки экспонентов некоторых множеств орграфов

Исследуем примитивность частного класса множеств орграфов и определим минимальное подмножество, на котором достигается оценка экспонента. Используем универсальный критерий примитивности орграфа [1]: орграф Γ примитивный, если и только если Γ сильносвязный и длины его простых контуров взаимно простые.

В дальнейшем потребуется следующее утверждение.

Утверждение 7. Пусть n-вершинный орграф Γ содержит гамильтонов контур $(0, \ldots, n-1)$ и множество дуг $E = \{(i, (i+l) \bmod n) : i = 0, \ldots, n-1\}$, $n \geqslant l > 1$. Тогда орграф Γ примитивный, если НОД$(n, l-1) = 1$, и $\exp \Gamma = n - 1$.

Доказательство. В данных условиях в сильносвязном орграфе Γ через каждую вершину проходит контур длины $(n-l+1)$. Так как НОД$(n, l-1) =$ НОД$(n, n-l+1) = 1$, орграф Γ примитивный согласно универсальному критерию.

В орграфе Γ при любых $i, k = 0, \ldots, n-1$ путь длины t из вершины i в вершину $j = (i + k) \bmod n$ состоит из x_k дуг множества E и $(t - x_k)$ дуг контура $(0, \ldots, n-1)$, где $t \geqslant x_k$. Следовательно, выполнена система сравнений

$$i + x_k l + t - x_k \equiv i + k \pmod{n}, \;\; k = 0, \ldots, n-1.$$

После элементарных преобразований данная система принимает вид

$$(l-1)x_k \equiv k - t \pmod{n}, \;\; k = 0, \ldots, n-1. \tag{2}$$

Все сравнения системы (2) при любых фиксированных k, t имеют единственное решение относительно x_k, так как НОД$(n, l-1) = 1$. Значит, при $k = 0, \ldots, n-1$ величина x_k принимает все значения от 0 до $n-1$. Длина t пути одинакова при любых $i, k = 0, \ldots, n-1$ и $t \geqslant x_k$. Следовательно, в орграфе Γ есть путь длины t из любой вершины в любую при всех $t \geqslant n-1$, отсюда $\exp \Gamma \leqslant n-1$.

Докажем (от противного), что для пар вершин $(i, (i+k) \bmod n)$, $i = 0, \ldots, n-1$, отсутствует путь длины $(n-2)$ при $k = ((n-1)(l-1)-2) \bmod n$. Действительно, если в Γ есть путь длины $t = n-2$ при указанном k, то сравнение (2) принимает вид

$$(l-1)x_k \equiv (n-1)(l-1) \pmod{n}.$$

Данное сравнение имеет единственное решение $x_k \equiv (n-1) \bmod n$, что противоречит условию $x_k \leqslant t = n-2$. Следовательно, $\exp \Gamma \geqslant n-1$. Объединяя оценки, получаем $\exp \Gamma = n-1$. ■

Следствие 2 (при $l = n$). Если орграф Γ имеет петли во всех вершинах гамильтонова контура, то $\exp \Gamma = n-1$.

Докажем критерий примитивности множества орграфов, обобщающего множество орграфов Виландта, и оценим его экспонент.

Теорема 3. Пусть $\hat{\Gamma} = \{\Gamma_0, \ldots, \Gamma_{n-1}\}$ — множество орграфов с вершинами $0, \ldots, n-1$, где орграф Γ_i имеет гамильтонов контур $(0, \ldots, n-1)$ и дугу $(i, (i+l) \bmod n)$, $n \geqslant l > 1$, $i = 0, \ldots, n-1$. Тогда множество орграфов $\hat{\Gamma}$ примитивное, если и только если НОД$(n, l-1) = 1$, при этом

$$n - 1 \leqslant \exp \hat{\Gamma} \leqslant 2n - 2.$$

Доказательство. Необходимость. Пусть множество орграфов $\hat{\Gamma}$ примитивное. Тогда некоторое слово $\Gamma = \Gamma_{w_1} \dots \Gamma_{w_t}$ длины $t > 0$ в алфавите $\hat{\Gamma}$ является полным графом, то есть любая пара вершин в орграфе Γ есть дуга. В силу правила умножения графов это означает, что в мультиграфе $\Gamma^{(p)}$ для любых $i, k \in \{0, \dots, n-1\}$ имеется путь $[i, (i+k) \bmod n] = (i_{w_0}, i_{w_1}, \dots, i_{w_t})$ длины t, где $i_{w_0} = i$, $i_{w_t} = (i+k) \bmod n$ и $i_{w_s} = (i_{w_{s-1}} + \xi_s^{(i,k)}) \bmod n$, $\xi_s^{(i,k)} \in \{1, l\}$, $s = 1, \dots, t$. Обозначим через $x^{(i,k)}$ частоту символа l в слове $(\xi_1^{(i,k)}, \dots, \xi_t^{(i,k)})$, тогда $(t - x^{(i,k)})$ есть частота символа 1 в том же слове. Из существования пути $(i_{w_0}, i_{w_1}, \dots, i_{w_t})$ длины t при любых $i, k \in \{0, \dots, n-1\}$ следует, что выполнена система сравнений

$$i + lx^{(i,k)} + t - x^{(i,k)} \equiv i + k \pmod{n}, \;\; k = 0, \dots, n-1.$$

После элементарных преобразований данная система принимает вид

$$(l-1)x^{(i,k)} \equiv k - t \pmod{n}, \;\; k = 0, \dots, n-1. \tag{3}$$

По построению каждое сравнение системы (3) при фиксированном t имеет решение относительно $x^{(i,k)}$, что возможно только при условии НОД$(n, l-1) = 1$.

Достаточность. При НОД$(n, l-1) = 1$ орграфы $\Gamma_0, \dots, \Gamma_{n-1}$ примитивные, так как каждый из них сильносвязный и содержит контуры длины n и $(n-l+1)$, где НОД$(n, n-l+1) = 1$. Следовательно, множество $\hat{\Gamma}$ примитивное. Для получения оценок экспонента множества $\hat{\Gamma}$ докажем промежуточную лемму.

Для $a \in \mathbb{N}$ и $Y \subset \mathbb{N}$ положим $(a \pm Y) \bmod n = \{(a \pm y) \bmod n : y \in Y\}$. Обозначим S_r, $r = 1, 2, \dots$, последовательность множеств чисел, где $S_1 = \{1, l\}$, $S_r = (1 + S_{r-1}) \bmod n \cup \cup \{(rl) \bmod n\}$, $r > 1$.

Лемма 1. При НОД$(n, l-1) = 1$ множество S_r содержит полную систему вычетов по модулю n, если и только если $r \geqslant n-1$.

Доказательство. Число a принадлежит S_r, если и только если $a = xl + (r - x)$, где $x \in \{0, \dots, r\}$. Следовательно, S_r содержит полную систему вычетов по модулю n, если и только если в множестве $\{0, \dots, r\}$ имеется решение (относительно x) каждого сравнения системы

$$xl + r - x \equiv a \pmod{n}, \quad a = 0, \dots, n-1.$$

После элементарных преобразований получаем, что указанная система сравнений равносильна следующей:

$$\{x(l-1) \equiv a - r \pmod{n}, \quad a = 0, \dots, n-1,$$

где функция $x(l-1) \bmod n$ при НОД$(n, l-1) = 1$ реализует подстановку множества $\{0, \dots, n-1\}$. Значит, каждое сравнение исходной системы имеет решение в множестве $\{0, \dots, r\}$, если и только если $r \geqslant n-1$. Лемма доказана. ■

Построим слово длины $(2n-2)$ в алфавите $\hat{\Gamma}$, которому при умножении соответствует полный орграф.

Обозначим $G_{r,1} = \Gamma_{(n-lr) \bmod n} \cdot \ldots \cdot \Gamma_{(n-l) \bmod n}$, $r = 1, \dots, n-1$. Покажем (индукция по r), что в орграфе $G_{r,1}$ есть дуги $(j, 0)$ для любого $j \in (n - S_r) \bmod n$.

При $r = 1$ выполнено: $S_1 = \{1, l\}$, $G_{1,1} = \Gamma_{n-l}$. Орграф Γ_{n-l} имеет дуги $(n-1, 0)$ и $(n-l, 0)$ по определению, то есть при $r = 1$ утверждение верно.

Пусть утверждение доказано для $r = 1, \ldots, \tau - 1$, где $1 < \tau \leqslant n - 1$, докажем его для $r = \tau$.

По предположению индукции, для любого $j \in (n - S_{\tau-1}) \bmod n$ в орграфе $G_{\tau-1,1}$ есть дуга $(j, 0)$. По условию $G_{\tau,1} = \Gamma_{(n-l\tau) \bmod n} \cdot G_{\tau-1,1}$, где орграф $\Gamma_{(n-l\tau) \bmod n}$ имеет гамильтонов контур $(0, \ldots, n-1)$ и дугу $((n - l\tau) \bmod n, (n - l(\tau - 1)) \bmod n)$. Тогда, в соответствии с правилом умножения орграфов, орграф $G_{\tau,1}$ имеет дуги $(j, 0)$ для любого $j \in (n - S_{\tau-1} - 1) \bmod n$, а также дугу $((n - l\tau) \bmod n, 0)$. По определению $S_\tau = (1 + S_{\tau-1}) \bmod n \cup \{(\tau l) \bmod n\}$, следовательно, орграф $G_{\tau,1}$ имеет дуги $(j, 0)$ для любого $j \in (n - S_\tau) \bmod n$.

По лемме 1 множество S_{n-1} содержит полную систему вычетов по модулю n. Следовательно, в орграфе $G_{n-1,1}$ есть дуги $(j, 0)$ при $j = 0, \ldots, n - 1$.

Обозначим $G_{0,r} = \Gamma_{0 \bmod n} \cdot \ldots \cdot \Gamma_{lr \bmod n}$, $r = 0, \ldots, n - 2$. Покажем (индукция по r), что в орграфе $G_{0,r}$ есть дуги $(0, j)$ для любого $j \in S_{r+1}$.

При $r = 0$ выполнено: $S_1 = \{1, l\}$, $G_{0,0} = \Gamma_0$. Орграф Γ_0 имеет дуги $(0, 1)$ и $(0, l)$ по определению, то есть при $r = 0$ утверждение верно.

Пусть утверждение доказано для $r = 0, \ldots, \tau - 1$, где $0 < \tau \leqslant n - 2$, докажем его для $r = \tau$.

По предположению индукции, для любого $j \in S_\tau$ в орграфе $G_{0,\tau-1}$ есть дуга $(0, j)$. По условию $G_{0,\tau} = G_{0,\tau-1} \cdot \Gamma_{l\tau \bmod n}$, где орграф $\Gamma_{l\tau \bmod n}$ имеет гамильтонов контур $(0, \ldots, n-1)$ и дугу $(l\tau \bmod n, (\tau + 1)l \bmod n)$. Тогда, в соответствии с правилом умножения орграфов, орграф $G_{0,\tau}$ имеет дуги $(0, j)$ для любого $j \in (S_\tau + 1) \bmod n$, а также дугу $(0, (\tau+1)l \bmod n)$. По определению $S_{\tau+1} = (1 + S_\tau) \bmod n \cup \{(\tau+1)l \bmod n\}$, значит, орграф $G_{0,\tau}$ имеет дуги $(0, j)$ для любого $j \in S_{\tau+1}$.

По лемме 1 множество S_{n-1} содержит полную систему вычетов по модулю n. Следовательно, в орграфе $G_{0,n-2}$ есть дуги $(0, j)$ при $j = 0, \ldots, n - 1$.

В соответствии с правилом умножения орграфов, в произведении $\Gamma = G_{n-1,1} \cdot G_{0,n-2}$ есть дуги из любой вершины в любую, т. е. орграф Γ полный и $\exp \Gamma \leqslant 2n - 2$.

Для получения нижней оценки экспонента множества рассмотрим орграф $\Gamma^{(n)} = \Gamma_0 \cup \ldots \cup \Gamma_{n-1}$. Если множество $\hat{\Gamma}$ примитивное, то орграф $\Gamma^{(n)}$ также примитивный [5, с. 188] и $\exp \hat{\Gamma} \geqslant \exp \Gamma^{(n)}$. По утверждению 7 $\exp \Gamma^{(n)} = n - 1$. Теорема 3 доказана. ■

Замечание 3. При $l = 2$ имеем множество графов Виландта. Тогда $\exp \Gamma_0 = \ldots = \exp \Gamma_{n-1} = n^2 - 2n + 2$, вместе с тем $\exp \hat{\Gamma} \leqslant 2n - 2$.

Замечание 4. При $l = n$ орграф Γ_i содержит гамильтонов контур и петлю в вершине i, тогда $\exp \hat{\Gamma} = \exp \Gamma_i = 2n - 2$, $i = 0, \ldots, n - 1$.

Замечание 5. Полугруппа $\langle \hat{\Gamma} \rangle$ содержит не менее n положительных слов длины $(2n - 2)$, а именно содержит слова $G^{(k)}_{n-1,1} \cdot G^{(k)}_{0,n-2}$ при $k = 0, \ldots, n - 1$, где $G^{(k)}_{n-1,1} = \Gamma_{(k-l(n-1)) \bmod n} \cdot \ldots \cdot \Gamma_{(k-l) \bmod n}$, $G^{(k)}_{0,n-2} = \Gamma_{k \bmod n} \cdot \ldots \cdot \Gamma_{(k+l(n-2)) \bmod n}$. Доказательство проводится аналогично.

Следствие 3. Минимальное примитивное подмножество множества орграфов $\hat{\Gamma} = \{\Gamma_0, \ldots, \Gamma_{n-1}\}$, на котором достигается верхняя оценка теоремы 3, является сокращённым и содержит не более n/d орграфов, где $d = \text{НОД}(n, l)$.

Доказательство. Множество $\{(n - lr) \bmod n : r = 1, \ldots, n - 1\} \cup \{lr \bmod n : r = 0, \ldots, n - 2\} = \{lr \bmod n : r = 0, \ldots, n - 1\}$ содержит n/d различных вычетов по модулю n, где $d = \text{НОД}(n, l)$. Следовательно, в произведении $\Gamma = G_{n-1,1} \cdot G_{0,n-2}$ записаны n/d различных орграфов.

Множество орграфов $\hat{\Gamma}$ сокращённое, так как ни один орграф из $\Gamma_0, \ldots, \Gamma_{n-1}$ не является частью другого орграфа. ■

Пример 2. Пусть $n = 4$, $\hat{\Gamma} = \{\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3\}$. При $l = 2$ НОД$(n, n - l + 1) =$ $=$ НОД$(4, 3) = 1$ (множество графов Виландта). Орграф Γ_0 содержит дугу $(0, 2)$, Γ_1 содержит дугу $(1, 3)$ и т. д. Тогда $G_{3,1} = \Gamma_2\Gamma_0\Gamma_2$, $G_{0,2} = \Gamma_0\Gamma_2\Gamma_0$. Орграф $G_{3,1}$ имеет дуги $(0, 0)$, $(1, 0)$, $(2, 0)$, $(3, 0)$; орграф $G_{0,2}$ имеет дуги $(0, 0)$, $(0, 1)$, $(0, 2)$, $(0, 3)$. Тогда $G_{3,1} \cdot G_{0,2}$ полный и $\exp\hat{\Gamma} \leqslant 6$.

Пример 3. Пусть $n = 5$, $\hat{\Gamma} = \{\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$. При $l = 3$ НОД$(n, n - l + 1) =$ $=$ НОД$(5, 3) = 1$. Орграф Γ_0 содержит дугу $(0, 3)$, Γ_1 содержит дугу $(1, 4)$ и т. д. Тогда $G_{4,1} = \Gamma_3\Gamma_1\Gamma_4\Gamma_2$, $G_{0,3} = \Gamma_0\Gamma_3\Gamma_1\Gamma_4$. Орграф $G_{4,1}$ имеет дуги $(0, 0)$, $(1, 0)$, $(2, 0)$, $(3, 0)$, $(4, 0)$; орграф $G_{0,3}$ имеет дуги $(0, 0)$, $(0, 1)$, $(0, 2)$, $(0, 3)$, $(0, 4)$. Тогда $G_{4,1} \cdot G_{0,3}$ полный и $\exp\hat{\Gamma} \leqslant 8$.

Выводы

1) Показано, что исследование примитивности данного множества неотрицательных матриц (орграфов) может быть сведено к исследованию примитивности подмножества. Задача минимизации нетривиальна для множеств, состоящих из непримитивных матриц (орграфов).
2) Получен универсальный критерий примитивности множества $\hat{M}$ неотрицательных матриц (множества $\hat{\Gamma}$ орграфов). В силу конечности полугруппы $\langle\hat{M}\rangle$ (полугруппы $\langle\hat{\Gamma}\rangle$) при фиксированном n задача распознавания примитивности множества матриц $\hat{M}$ (орграфов $\hat{\Gamma}$) алгоритмически разрешима. Актуальной задачей является построение соответствующего алгоритма и оценка его вычислительной сложности.
3) Показано, что критерий примитивности множества перемешивающих орграфов может быть упрощен для частных классов, в том числе для классов, важных с прикладной точки зрения.
4) На примере некоторых множеств примитивных орграфов показано, что экспонент множества может быть существенно меньше экспонента любого орграфа из данного множества.

ЛИТЕРАТУРА

1. *Сачков В. Н., Тараканов В. Е.* Комбинаторика неотрицательных матриц. М.: ТВП, 2000. 448 с.
2. *Князев А. В.* Оценки экстремальных значений основных метрических характеристик псевдосимметрических графов: дис. ... д-ра физ.-мат. наук. М., 2002. 203 с.
3. *Фомичев В. М.* Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010. 424 с.
4. *Авезова Я. Э., Фомичев В. М.* Комбинаторные свойства систем разноразмерных 0, 1-матриц // Прикладная дискретная математика. 2014. № 2(24). С. 5–11.
5. *Фомичев В. М., Мельников Д. А.* Криптографические методы защиты информации. Ч. 1. Математические аспекты. М.: Изд-во Юрайт, 2016. 209 с.
6. *Кяжин С. Н., Фомичев В. М.* Локальная примитивность графов и неотрицательных матриц // Прикладная дискретная математика. 2014. № 3(25). С. 68–80.
7. *Берж К.* Теория графов и её применение. М.: ИЛ, 1962. 320 с.
8. *Фомичев В. М., Кяжин С. Н.* Локальная примитивность матриц и графов // Дискретный анализ и исследование операций. 2017. Т. 24. № 1. С. 97–119.

9. *Фомичев В. М.* Новая универсальная оценка экспонентов графов // Прикладная дискретная математика. 2016. № 3(33). С. 78–84.

REFERENCES

1. *Sachkov V. N. and Tarakanov V. E.* Kombinatorika neotritsatel'nykh matrits [Combinatorics of Non-Negative Matrices]. Moscow, TVP Publ., 2000, 448 p. (in Russian)
2. *Knyazev A. V.* Otsenki ekstremal'nykh znacheniy osnovnykh metricheskikh kharakteristik psevdosimmetricheskikh grafov [Estimates for the Extreme Values of the Basic Metric Characteristics of Pseudosymmetric Graphs]. Doctor of Physics and Mathematics Thesis, Moscow, 2002, 203 p. (in Russian)
3. *Fomichev V. M.* Metody diskretnoy matematiki v kriptologii [Methods of Discrete Mathematics in Cryptology]. Moscow, Dialog-MEPhI Publ., 2010, 424 p. (in Russian)
4. *Avezova Ya. E. and Fomichev V. M.* Kombinatornye svoystva sistem raznorazmernykh 0, 1-matrits [Combinatorial properties of rectangular 0,1-matrix systems]. Prikladnaya Diskretnaya Matematika, 2014, no. 2(24), pp. 5–11. (in Russian)
5. *Fomichev V. M. and Mel'nikov D. A.* Kriptograficheskie metody zashchity informatsii. Ch. 1. Matematicheskie aspekty [Cryptographic Methods of Information Security. Part 1. Mathematical Aspects]. Moscow, Yurayt Publ., 2016, 209 p. (in Russian)
6. *Kyazhin S. N. and Fomichev V. M.* Lokal'naya primitivnost' grafov i neotritsatel'nykh matrits [Local primitiveness of graphs and nonnegative matrices]. Prikladnaya Diskretnaya Matematika, 2014, no. 3(25), pp. 68–80. (in Russian)
7. *Berzh K.* Teoriya grafov i ee primenenie [Graph Theory and its Application.]. Moscow, Foreign Literature Publ., 1962, 320 p. (in Russian)
8. *Fomichev V. M. and Kyazhin S. N.* Lokal'naya primitivnost' matrits i grafov [Local primitiveness of matrices and graphs]. Diskretn. Anal. Issled. Oper., 2017, vol. 24, no. 1, pp. 97–119. (in Russian)
9. *Fomichev V. M.* Novaya universal'naya otsenka eksponentov grafov [The new universal estimation for exponents of graphs]. Prikladnaya Diskretnaya Matematika, 2016, no. 3(33), pp. 78–84. (in Russian)

ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

УДК 681.3.06: 681.323

КЛЕТОЧНО-АВТОМАТНЫЕ МОДЕЛИ ЕСТЕСТВЕННЫХ ПРОЦЕССОВ И ИХ РЕАЛИЗАЦИЯ НА СОВРЕМЕННЫХ КОМПЬЮТЕРАХ

О. Л. Бандман

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

Представлены результаты анализа моделирующих способностей и вычислительных свойств методов клеточно-автоматного (КА) моделирования нелинейных пространственно распределенных процессов. Работа преследует две цели: 1) показать соответствие свойств КА-моделей современным тенденциям развития параллельных многопроцессорных архитектур (дискретность представления данных, локальность взаимодействий) и 2) раскрыть возможности КА-методов для компьютерного моделирования естественных существенно нелинейных, диссипативных процессов, не поддающихся традиционным методам математического моделирования. Обобщается опыт КА-моделирования, полученный в ИВМиМГ СО РАН. Работа содержит формальное представление КА-моделей, методы их построения, а также результаты реализации ряда тестовых и реальных задач на суперкомпьютерах.

Ключевые слова: *математическое моделирование, дискретная математика, параллельные вычисления, клеточно-автоматные модели пространственной динамики, клеточно-автоматная гидродинамика, реакционно-диффузионные процессы.*

DOI 10.17223/20710410/35/9

CELLULAR-AUTOMATA MODELS OF NATURAL PROCESSES, IMPLEMENTATION ON SUPERCOMPUTERS

O. L. Bandman

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

E-mail: bandman@ssd.sscc.ru

Conventional mathematical models based on differential calculus are sometimes not capable to simulate nonlinear dissipative processes on micro- or nano-level of resolution. This fact stimulates the development of new approaches to spatial dynamics simulation. Among them, cellular automata (CA) modeling is one of the promising methodologies, due to CA large simulation capability and compatibility with modern trends in supercomputer architecture. Although CA simulation is intensively studied and used in different fields, a few attention is paid to studying the parallel implementation peculiarities of large scale CA-models on supercomputers. Just this aspect of

CA-simulation is the subject of the paper aiming to analyse CA-simulation methods adaptiveness to supercomputing implementation based on validity conditions requirements for different modes of CA operation. For this purpose, the concept of the operation mode well known for simple CA (having only one transition rule) is expanded for composed CA (containing many transition rules). The new concept determines a CA-transition rules execution order, which in turn determines the behavioral properties of CA-model and their influence on the simulation performance. The obtained results are illustrated by some examples, which show CA methods at work by simulating essentially nonlinear and dissipative processes: superposition of asynchronous CAs for simulation of water permeating through porous medium and parallel composition of two CAs simulating pattern formation on a heated plate. Basically, the paper generalizes CA computer simulation theoretical results and experience obtained by researchers from the Institute of Computational Mathematics and Mathematical Geophysics of Siberian Branch of Russian Academy of Sciences.

Keywords: *mathematical modeling, parallel computing, cellular automata models, Lattice-Gas hydrodynamics, reaction-diffusion processes.*

Введение

Клеточный автомат, введённый фон Нейманом в середине прошлого века [1], в настоящее время рассматривается как математическая модель пространственно распределённого естественного процесса, в которой все компоненты (пространство, время и состояния) дискретны. Теория клеточных автоматов в её современном виде [2] составляет часть теории динамических систем. Созданная Пуанкаре в позапрошлом веке теория динамических систем опиралась на математические описания явлений в виде дифференциальных уравнений. Появление компьютеров и численных методов распространило её положения на конечно-разностные представления [3]. Наконец, развитие КА-моделирования привело к появлению раздела, изучающего полностью дискретные динамические системы [4]. Однако понятие КА возникло не в результате постепенной эволюции теории динамических систем к дискретному виду, а в результате развития идеи фон Неймана о возможности построения вычислительной модели устройства, воспроизводящего себе подобного. Это создавало впечатление о КА как об абстрактном представлении некоторого биологического сообщества, состоящего из множества «клеток», образующих регулярную пространственную структуру. Каждая клетка является элементарным вычислителем, который может находиться в одном из двух состояний — 0 и 1 (белое или черное) и изменять это состояние в зависимости от состояний клеток ближайшего своего окружения, называемого соседством. Алгоритм вычисления следующего состояния у всех клеток одинаковый. Все клетки выполняют переход в новое состояние одновременно, изменяя таким образом глобальную черно-белую картину распределения состояний по пространству.

Интерпретация и осмысливание концепции КА происходили вместе с развитием вычислительной науки и технологии и в эпоху современных суперкопьютеров приняли форму нового направления в математическом и компьютерном моделировании. Это вызвано тем, что стало ощущаться несоответствие между идеологией, на которой основана работа компьютера, и идеологией, на которой построена вычислительная математика. Противоречия между ними сводятся к следующим двум положениям.

1. Компьютер работает с дискретными данными в дискретном времени, тогда как вычислительные модели основаны на непрерывной математике. Это противоречие имеет два последствия. Во-первых, при переходе к вычислительным алгоритмам диффе-

ренциальные уравнения аппроксимируются разностными схемами, что приводит к потере точности. Во-вторых, непрерывность обрабатываемых данных требует работы с вещественными числами, т. е. использования арифметики с плавающей точкой, которая отбрасывает малые разряды в процессе счёта. Это приводит к неточным результатам и, что ещё более важно, вызывает неустойчивость вычислительного процесса. Последнее является сильным ограничением для выбора шага по времени при использовании явных схем дискретизации.

2. Тенденции развития современных суперкомпьютеров направлены в сторону увеличения количества параллельно работающих процессоров, тогда как вычислительная математика остаётся на фундаменте последовательных алгоритмов. Возникает проблема распараллеливания, которая решается не всегда эффективно.

Поиск новых моделей, свободных от этих противоречий, заставил пересмотреть концепцию КА. Новый взгляд на КА как на модель пространственной динамики, изложенный в фундаментальных работах [5, 6], привёл к осознанию того факта, что, несмотря на простоту каждой клетки, их кооперативная работа может моделировать сложные и разнообразные процессы, которые иногда невозможно (или, по крайней мере, неизвестно как) описать другим способом. И, что немаловажно, имея одного и того же «родителя», КА и компьютер близки по «духу и букве», что улучшает их совместимость. Настоящим потрясением основ в моделировании пространственной динамики стало появление клеточно-автоматной гидродинамики, названной «Lattice-Gas Cellular Automata» (LGA — модель вязкой жидкости), для которой была доказана эквивалентность уравнению Навье — Стокса [7]. Вслед за LGA появились КА-модели процессов диффузии [6], разделения фаз [8], реакционно-диффузионных процессов [9, 10], знаменитой реакции Белоусова — Жаботинского [11], движения солитонов [12] и др. Теория и методология КА-моделирования интенсивно развиваются: проводятся ежегодно 3–4 международных конференции, выпускаются два специальных международных журнала («Complex Systems», США, и «Cellular Automata», Великобритания). В результате оформилось новое направление компьютерного моделирования, в котором понятие «клеточного автомата» фон Неймана играет роль идеологической основы. К сожалению, в России только отдельные разрозненные группы занимаются КА-моделированием.

Цель работы — показать моделирующие способности КА-методов и их адаптируемость к современной параллельной вычислительной технике, иллюстрируя их результатами, полученными в течение ряда лет группой исследователей Института вычислительной математики и математической геофизики СО РАН. В п. 1 даны формальные определения основных понятий и классификация КА-моделей; п. 2 посвящён КА-моделям гидродинамики и сопоставлению их с непрерывными аналогами. В п. 4 представлены композиции КА, моделирующие сложные процессы. В заключении формулируются выводы из изложенного и проблемы дальнейшего развития.

1. Клеточно-автоматные модели пространственной динамики

1.1. Формальное представление клеточно-автоматной модели

Формально, КА-модель определяется четвёркой понятий [13, 14]

$$\aleph = \langle A, X, \Theta, \rho \rangle,$$

где X — множество клеток, представленных своими именами; A — алфавит состояний клеток в X; Θ — множество локальных операторов клеток, называемое также глобаль-

ным оператором; ρ — режим функционирования модели. Состояниями клеток могут быть числа, булевы векторы или символы, именами клеток — натуральные числа или векторы координат точек дискретного пространства. Клетка $x \in X$ в состоянии $a \in A$ записывается как пара (a, x). Таким образом, $A \times X$ есть множество всех клеток во всевозможных их состояниях. Любое его подмножество $\Omega = \{(a_1, x_1), \ldots, (a_s, x_s)\}$, где $\{x_1, \ldots, x_s\} = X$, называется клеточным массивом КА-модели, а набор состояний клеток в нём $(a_1, \ldots, a_s)$ — его и КА-модели (глобальным) состоянием.

Для определения локального оператора вводятся именующие функции $\varphi_i : X \to X$, $i = 1, 2, \ldots, q$, такие, что для каждого $x \in X$ все клетки $x, \varphi_1(x), \ldots, \varphi_q(x)$ различные. Их подмножества

$$N(x) = \{x, \varphi_1(x), \ldots, \varphi_n(x)\}, \quad E(x) = \{\varphi_{n+1}(x), \ldots, \varphi_q(x)\}, \quad 0 \leqslant n \leqslant q,$$

называются соответственно соседством и окрестностью клетки x, а их подмножества вместе с некоторыми их состояниями

$$S(x) = \{(u_0, x), (u_1, \varphi_1(x)), \ldots, (u_n, \varphi_n(x))\}, \quad C(x) = \{(u_{n+1}, \varphi_{n+1}(x)), \ldots, (u_q, \varphi_q(x))\}$$

— соответственно локальной конфигурацией и контекстом для этой клетки. Множества всевозможных локальных конфигураций и контекстов для клетки x обозначаются $\mathcal{S}(x)$ и $\mathcal{C}(x)$ соответственно. Таким образом,

$$\mathcal{S}(x) = \{S(x) = \{(u_0, x), (u_1, \varphi_1(x)), \ldots, (u_n, \varphi_n(x))\} : u_0, u_1, \ldots, u_n \in A\},$$
$$\mathcal{C}(x) = \{C(x) = \{(u_{n+1}, \varphi_{n+1}(x)), \ldots, (u_q, \varphi_q(x))\} : u_{n+1}, \ldots, u_q \in A\}.$$

Локальные операторы в Θ ставятся в соответствие клеткам так, что каждой клетке соответствуют один или более таких операторов. На множестве $\Theta(x)$ операторов, сопоставленных клетке x, может быть задано распределение вероятности, в соответствии с которым может определяться порядок их выбора для применения в процессе функционирования КА-модели в том или ином режиме. Произвольный локальный оператор θ в $\Theta(x)$ обозначается $\theta(x)$ и называется оператором клетки x. Это есть отображение $\theta(x) : \mathcal{S}(x) \times \mathcal{C}(x) \to \mathcal{S}(x)$, определяемое следующим образом: результат $S'(x) = \theta(x)[S(x), C(x)]$ его действия над произвольной парой $[S(x), C(x)] \in \mathcal{S}(x) \times \mathcal{C}(x)$ получается преобразованием локальной конфигурации $S(x)$ клетки x к её новой локальной конфигурации

$$S'(x) = \theta(x)[S(x), C(x)] = \{(u'_0, x), (u'_1, \varphi_1(x)), \ldots, (u'_n, \varphi_n(x))\}$$

путём замены состояний $u_0, u_1, \ldots, u_n$ клеток соседства $N(x)$ новыми состояниями $u'_0, u'_1, \ldots, u'_n$ соответственно, вычисленными как

$$u'_k = f_k(u_0, u_1, \ldots, u_q), \quad k = 0, 1, \ldots, n,$$

где $f_k : A^{q+1} \to A$ и векторная функция $f = (f_0 f_1 \ldots f_q)$ называется функцией переходов.

Далее в случае пустого контекста, т. е. когда $q = n$, вместо $\theta(x)[S(x), C(x)]$ пишем $\theta(x)[S(x)]$. Кроме того, локальную конфигурацию $S(x)$ и контекст $C(x)$ под знаком оператора $\theta(x)$ будем относить не только к клетке x, но и к оператору $\theta(x)$, называя их его локальной конфигурацией и контекстом соответственно.

Нетрудно заметить, что данное определение локального оператора $\theta(x)$ фактически совпадает с определением конечного автомата с входным алфавитом A^q, множеством

состояний A, выходным алфавитом A^n и функциями переходов $\psi : A^q \times A \to A$ и выходов $\varphi : A^q \times A \to A^n$, удовлетворяющими соотношениям $\psi(u_1u_2 \dots u_nu_{n+1} \dots u_q, u_0) = u'_0$ и $\varphi(u_1u_2 \dots u_nu_{n+1} \dots u_q, u_0) = u'_1u'_2 \dots u'_n$, где u_0 и u'_0 являются соответственно текущим и следующим состояниями клетки x; $u_1, u_2, \dots, u_n$ и $u'_1, u'_2, \dots, u'_n$ — соответственно текущие и новые состояния соседних для x клеток; $u_{n+1}, u_{n+2}, \dots, u_q$ — текущие состояния клеток из контекста клетки x. Именно поэтому рассматриваемые здесь модели реальных процессов носят название клеточно-автоматных.

При заданном начальном клеточном массиве $\Omega(0)$ функционирование КА-модели $\aleph$ происходит в дискретном времени $t = 0, 1, \dots$, каждый промежуток которого называется итерацией. Результатом t-й итерации является клеточный массив $\Omega(t+1) = \Theta(\Omega(t))$. Он получается применением локальных операторов $\theta(x) \in \Theta$ ко всем клеткам $(u, x) \in \Omega(t)$ в порядке, определяемом заданным режимом, и тем самым совершается глобальный переход $\Omega(t) \to \Omega(t+1)$ от текущего клеточного массива к следующему. Применение $\theta(x)$ к клетке $(u, x) \in \Omega(t)$ означает замену состояний клеток из $S(x)$ в $\Omega(t)$ на состояния одноименных клеток из $S'(x) = \theta(x)[S(x), C(x)]$.

Здесь уместно обратить внимание на роли локальной конфигурации $S(x)$ и контекста $C(x)$ как частей клеточного массива $\Omega(t)$ в определении локального оператора $\theta(x)$. Вместе они образуют аргументы, по значениям которых вычисляется значение $S'(x)$ на выходе $\theta(x)$. С другой стороны, в результате выполнения оператора значение контекста $C(x)$ не изменяется, а на место конфигурации $S(x)$ записывется новая — $S'(x)$.

Говорят, что глобальные переходы в КА-модели совершаются корректно [15], а вместе с ними корректна и КА-модель, если в процессе вычислений $\Omega(t+1)$ по $\Omega(t)$ для всех $t \geqslant 0$ не происходит ни одной коллизии, т. е. ни одной попытки изменить состояние одной и той же клетки более одного раза в один и тот же момент времени t. Формально это условие выражается так: для любых $\theta_k(x), \theta_l(y) \in \Theta$, $x, y \in X$ и $S(x) \in \mathcal{S}(x)$, $S(y) \in \mathcal{S}(y)$

$$x \neq y \Rightarrow (N(\theta_k(x)[S(x), C(x)]) \cap N(\theta_l(y)[S(y), C(y)]) = \varnothing), \tag{1}$$

где $N(S'(x)) = \{x, \varphi_1(x), \dots, \varphi_n(x)\}$.

Результат итеративного применения глобального оператора Θ к массивам $\Omega(t)$ описывает эволюцию моделируемой системы в форме последовательности клеточных массивов $\Omega(0), \Omega(1) \dots, \Omega(t), \dots$ Если этот итеративный процесс сходится, т. е. существует t, такое, что $\Omega(t) = \Omega(t+1) = \dots$, то система приходит к устойчивому состоянию. В иных случаях КА моделирует колебательный или хаотический процесс [5].

Реальные процессы, как природные, так и рукотворные, обычно состоят из множества различных элементарных действий. Соответственно в их КА-моделях используется много разных локальных операторов [14]. В их обозначениях буквой θ последняя в рамках одной модели помечается далее разными числовыми индексами, как это сделано в (1). Для указания типа КА-модели, к которой относится данный оператор, или режима, в котором он применяется в ней, эта буква помечается символами, указывающими на эти тип модели или режим. Аналогичным образом метятся и буквы, обозначающие разные глобальные операторы и клеточные массивы в рассматриваемых моделях.

Различают три уровня сложности КА-моделей:

1) простые КА-модели, имитирующие одно элементарное действие (диффузия, конвекция, фазовые переходы, химические реакции и т. д.), используют один локальный оператор, $|\Theta| = 1$;

2) сложные КА-модели, в них глобальный оператор работает с множеством локальных операторов, $|\Theta| > 1$;
3) композиции глобальных операторов Θ_i, такие, как последовательная (суперпозиция глобальных операторов) и параллельная (система глобальных операторов). В них все Θ_i работают последовательно или параллельно, обновляя предписанные им подмассивы $\Omega_i \subset \Omega$ и используя в качестве контекстов наборы клеток из всего массива Ω.

Во всех случаях итерацией считается исполнение глобального оператора $\Theta(\Omega(t))$, т. е. процедура, состоящая из $|X| \cdot |\Theta|$ шагов применения всех $\theta_i(x) \in \Theta(x)$ ко всем $(u, x) \in \Omega(t)$. Порядок исполнения этих шагов определяется режимом функционирования ρ, который, в случае сложной модели, имеет две составляющие: пространственную ρ_x, определяющую порядок выбора клеток, и операторную ρ_θ, определяющую порядок выбора локальных операторов. В случае композиций глобальных операторов каждый оператор работает в своем режиме, а взаимодействие между ними определяется типом их композиции: последовательная или параллельная.

1.2. Режимы функционирования

Режим работы простой КА-модели характеризуется только пространственной составляющей ρ_x, обозначаемой индексами из множества греческих символов $\{\sigma, \alpha, \varkappa, \beta\}$, которыми индексируются операторы, исполняемые в соответствии со следующими алгоритмами.

- В синхронном режиме (режим σ) применение $\theta_\sigma(x)$ ко всем клеткам $(u, x) \in \Omega(t)$ происходит в любом порядке. Новые значения (u', x) записываются во вспомогательный массив Ω', который после заполнения играет роль $\Omega(t+1)$. Для синхронных КА-моделей $\aleph_\sigma$ условие (1) обеспечивается применением к каждой клетке x операторов $\theta_\sigma(x)$, не изменяющих состояний других клеток, что в классических синхронных КА выполняется по определению. При синтезе КА-модели этот факт является ограничением, зато при реализации синхронной КА-модели на многих процессорах параллельно обмен состояниями граничных клеток производится один раз за итерацию, что обычно гарантирует высокую эффективность параллельной реализации.
- В асинхронных режимах операторы $\theta_\alpha(x)$ или $\theta_\varkappa(x)$ применяются к клеткам $(u, x) \in \Omega(t)$, выбираемым последовательно в случайном (режим α) или заданном (режим $\varkappa$) порядке, причём текущие состояния клеток $(u_j, \varphi_j(x)) \in S(x)$ заменяются на новые $(u'_j, \varphi_j(x)) \in S'(x)$ немедленно. При реализации на одном процессоре условие (1) выполняется автоматически, независимо от количества клеток в $S(x)$, изменяющих свои состояния одновременно. Однако при параллельной реализации на кластере асинхронный режим требует выполнения межпроцессорных обменов после обновления состояния каждой клетки, имеющей соседа на другом процессоре. Это делает процесс распараллеливания практически бессмысленным. Решение проблемы состоит в преобразовании асинхронного режима в блочно-синхронный (режим β), что, как показано на ряде примеров [16, 17], не искажает результатов моделирования, зато обеспечивает приемлемую эффективность параллельной реализации.
- Блочно-синхронный (режим β) [16] является промежуточным между асинхронным и синхронным. На множестве X строится разбиение $\Pi(X) = \{X_1, X_2, \ldots, X_m\}$, такое, что

$$\begin{array}{rl} m & \geqslant |N(S'(x))|, \\ \forall X_k \in \Pi(X) \forall x, y \in X_k \quad [(x \neq y) & \Rightarrow (N(S'(x)) \cap N(S'(y)) = \varnothing)]. \end{array} \tag{2}$$

Итерация разбивается на m стадий, исполняемых последовательно в любом (возможно, случайном) порядке, причем на k-й стадии операторы $\theta_\beta(x)$ применяются синхронно ко всем клеткам с именами в X_k, $k = 1, 2, \ldots, m$. Режим позволяет преобразовывать локальные конфигурации любого размера $n > 1$, отсутствие коллизий при этом гарантируется условиями (2). При реализации Θ_β на параллельных системах требуется m межпроцессорных обменов за итерацию, что вполне приемлемо по эффективности.

Режим функционирования сложной модели характеризуется не только пространственной, но и операторной составляющей и обозначается парой символов (ρ_x, ρ_θ), причём $\rho_\theta \in \{\delta, \gamma\}$ и указывает на один из двух разных способов назначения оператора $\theta(x) \in \Theta(x)$ для применения к выбранному $x \in X$: детерминированный (режим δ) и вероятностный (режим γ). Наиболее известны следующие режимы сложных КА-моделей:

- Стохастический режим ($\rho = (\alpha, \gamma)$). Он является максимально рандомизированным представлением поведения частиц на микроуровне. Глобальный оператор $\Theta_{\alpha,\gamma}$ выполняется следующим образом:
  1) клетка $(u, x) \in \Omega(t)$, выбирается случайно (режим α);
  2) локальный оператор $\theta(x)$ выбирается среди операторов в $\Theta(x)$ по распределению частоты имитируемых ими действий и применяется к (u, x) немедленно.

  Итерация состоит из $|X| \cdot |\Theta|$ повторений п. 1 и 2. Стохастический режим используется в моделировании сложных химических реакций [17, 18].
- Блочно-синхронный детерминированный ($\rho = (\delta, \beta)$). Разбиение X на подмножества $X_1, \ldots, X_m$ выполняется согласно (2), с той лишь разницей, что
$$m \geqslant |\bigcup_{j=1}^{n} N(\theta_j(x)[S(x), C(x)])|.$$
  Итерация разбивается на m стадий, которые выбираются в детерминированном порядке, на k-й стадии все локальные операторы применяются в режиме (σ, δ).
- Блочно-синхронный стохастический режим ($\rho = (\alpha, \beta)$). Отличается от режима (δ, β) тем, что стадии выбираются в случайном порядке. Применяется при параллельных реализациях КА-моделей, изначально заданных в стохастическом режиме (α, γ).

Композиция глобальных операторов объединяет в единую модель несколько глобальных операторов, каждый из которых может быть простым или сложным и работать в своём заданном заранее режиме. Наиболее известны два типа композиций:

- Последовательная композиция глобальных операторов $\varphi(\Theta_1, \ldots, \Theta_L\}$, работающих каждый в своем режиме, причём исходным клеточным массивом для Θ_j служит $\Omega_{j-1}(t) = \Theta_{j-1}(\Omega(t))$. Итерация состоит из вычислений L промежуточных глобальных переходов:
$$\Omega(t+1) = \Theta_L(\Theta_{L-1}(\ldots(\Theta_1(\Omega(t))))).$$
  При реализации на суперкомпьютере последовательную композицию целесообразно размещать на кластерах, декомпозируя клеточный массив на подмассивы и организуя обмен состояниями их граничных клеток.

- В параллельной композиции [19] $\Psi(\Theta^{(1)},\ldots,\Theta^{(L)})$ множество клеток разделено на L равномощных подмножеств (слоев): $X = X^{(1)} \cup \ldots \cup X^{(L)}$, $X^{(j)} = \{x_i^{(j)} : i = 1,2,\ldots,N\}$, $j = 1,2,\ldots,L$, $|X| = N \cdot L$. Соответственно разделено и множество глобальных операторов: $\Omega = \{\Omega^{(1)} \cup \ldots \cup \Omega^{(L)}\}$. Все подмножества $\Omega^{(j)} \in \Omega$ обновляются одновременно соответствующими глобальными операторами $\Theta^{(j)}$, при этом контексты локальных операторов $\theta(i,j) \in \Theta$ могут быть в любых других подмножествах из Ω.

При реализации КА-модели на суперкомпьютере глобальные операторы целесообразно размещать на ядрах компьютера с общей памятью, где каждый КА ведёт вычисления на своем ядре, пользуясь состояниями клеток всех КА из общей памяти [20].

1.3. Классификация КА-моделей

Основными характеристиками всех КА-моделей, унаследованными от классического КА, являются дискретность пространства, времени и состояний, локальность межклеточных взаимодействий и итеративность глобальных переходов. Однако такие параметры КА-моделей, как алфавит состояний клеток, режим функционирования, тип и размер соседства, выбираются в соответствии со свойствами моделируемого явления и могут быть разного типа. На рис. 1 схематически показаны классы совре-

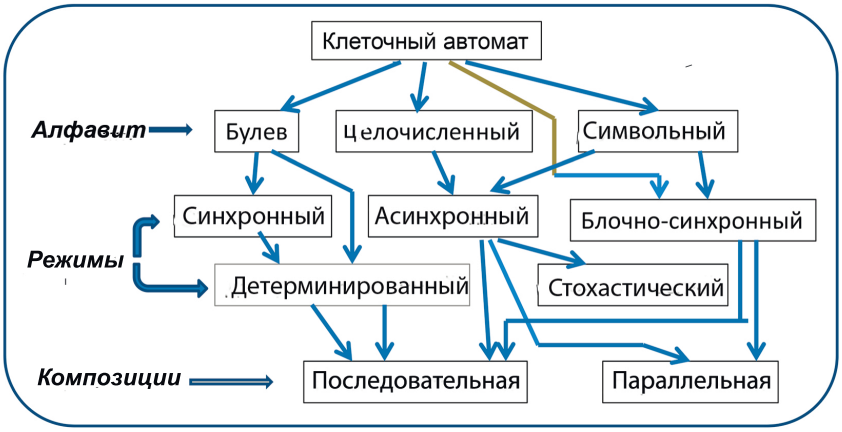

Рис. 1. Классы современных КА-моделей

менных КА-моделей [14]. Каждый исходящий из вершины путь к нижнему уровню соответствует определённому классу приложений. Классический КА [8] соответствует пути КА → булев → синхронный → детерминированный; КА-модели гетерогенных каталитических реакций [17, 18] лежат на пути КА → символьный → асинхронный → стохастический; целочисленная LGA-модель газового потока [21] соответствует пути КА → целочисленный → асинхронный → последовательная композиция.

С точки зрения методологии моделирования пространственной динамики КА-модели целесообразно подразделить на две группы: КА-модели, имеющие аналоги в традиционной (непрерывной) математике, и КА-модели, для которых таких аналогов нет. Представителями первой группы являются КА-модели газовых и жидкостных потоков (Lattice-Gas Cellular Automata [5, 7, 22]). Представителями второй — многочисленные КА-модели химических, биологических, социальных явлений, фазовых переходов.

2. КА-модели, имеющие аналоги в непрерывной математике

2.1. КА – модели диффузии

Самой востребованной простой КА-моделью, имеющей аналог в виде дифференциального уравнения, является КА-модель диффузии. Её популярность объясняется тем, что она присутствует как компонент в процессах типа «реакция — диффузия». Она существует в нескольких видах: асинхронный [6] и синхронный [23] варианты, причём каждая модель имеет булеву и целочисленную версии. Асинхронный двумерный КА, называемый «наивной диффузией», — это простейший КА, в котором $A = \{0, 1\}$; $X = \{(i, j) : i = 0, 1 \ldots, I; j = 0, 1, \ldots, J\}$; локальные операторы

$$\theta_l(i,j)[\{(u,(i,j)),(v_k,\varphi_k(i,j))\}] = \{(v_k,(i,j)),(u,\varphi_k(i,j))\}, \quad k = 0,1,2,3, \tag{3}$$

где в каждом применении оператора $\theta_l(i, j)$ соседняя клетка $\varphi_k(i, j)$ выбирается с вероятностью $p_d(i, j)$ в интервале $k/4 \leqslant p_d(i, j) < (k + 1)/4$, $k = 0, 1, 2, 3$. КА моделирует процесс изотропной диффузии с постоянным безразмерным коэффициентом диффузии $D = 0{,}5$ [24].

2.2. КА-модели потока вязкой жидкости

Известен ряд моделей газовых и жидкостных потоков, базовой моделью для которых служит КА-модель $\aleph = \langle A, X, \Theta, (\sigma, \delta)\rangle$, получившая название FHP-модели (по первым буквам фамилий авторов [7]). Модель является представлением кинетического процесса движения и столкновения абстрактных частиц в гексагональном дискретном пространстве. Расстояния между центрами соседних гексагонов (клеток) равны единице (рис. 2). Частицы могут перемещаться между центрами клеток с единичной скоростью. При столкновениях частицы меняют направления своего движения так, чтобы законы сохранения массы и импульса не нарушались (рис. 3). Множество имён клеток $X = \{x : x = (i, j), i = 0, \ldots I, j = 0, \ldots J\}$. Соседство клетки x содержит, кроме неё самой, еще шесть её ближайших соседей $N(x) = \{x, \varphi_1(x), \ldots, \varphi_6(x)\}$ (рис. 2). Алфавит состояний есть $A = \{w_k : k = 1, \ldots, 2^6\}$, где $w_k = (u_1, \ldots, u_6)$, $u_i \in \{0, 1\}$, $i = 1, \ldots, 6$. Если $u_i = 1$, то клетка $(w_k, x) \in \Omega$ содержит частицу, единичный вектор скорости $\mathbf{e}_i$ которой направлен к центру клетки $\varphi_i(x) \in N(x)$.

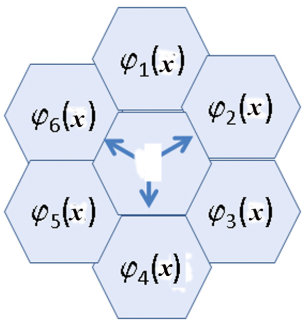

Рис. 2. Клетка (w, x), $w = (010101)$, и её шесть соседей

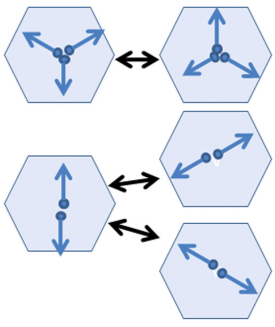

Рис. 3. Две функции столкновения: трёхчастичная детерминированная (вверху) и двухчастичная вероятностная (внизу)

Глобальный оператор перехода $\Theta^{\sigma,\delta}(\Omega)$ работает в режиме детерминированной суперпозиции синхронных глобальных операторов — сдвига $\Theta^{\sigma}_{M}(\Omega)$ и столкновения

$\Theta_C^\sigma(\Omega)$. Оператор сдвига применяется ко всем $(w_k, x) \in \Omega(t)$ синхронно. Он перемещает каждую частицу на одну клетку в направлении вектора её скорости. Это может быть выражено посредством локального оператора как

$$\theta_M(x)[\{(w,x)\},\{(w_1'',\varphi_1(x)),\ldots,(w_6'',\varphi_6(x))\}] = \{(w',x)\}, \tag{4}$$

где компоненты вектора нового состояния $w' = (u_1', \ldots, u_j', \ldots, u_6')$ равны компонентам векторов состояний контекстных для x клеток $w_i'' = (v_1^{(i)}, \ldots, v_j^{(i)}, \ldots, v_6^{(i)})$ со сдвигом, сохраняющим направление движения соответствующей частицы, т. е. $u_j' = v_j^{(\varphi(j+2) \bmod 6+1)}$.

Глобальный оператор $\Theta_C^\sigma(\Omega)$ состоит из одного бесконтекстного локального оператора $\theta_C(x)$, заменяющего состояние одной клетки x на значение вероятностной функции перехода, которая меняет направления векторов движения частиц, имитируя их столкновение:

$$\theta_C(x)[\{(u_1 \ldots u_6, x)\}] = \{(u_1' \ldots u_6', x)\}, \quad (u_1' \ldots u_6') = f(u_1 \ldots u_6). \tag{5}$$

Функция перехода f удовлетворяет законам сохранения массы и импульса:

$$\sum_{i=1}^{6} u_i = \sum_{i=1}^{6} u_i', \qquad \sum_{i=1}^{6} u_i \mathbf{e}_i = \sum_{i=1}^{6} u_i' \mathbf{e}_i. \tag{6}$$

В трёхмерном случае в [22] предложено использовать дискретное пространство, состоящее из ромбододэкаэдров, имеющих 12 граней и, следовательно, соседство из 12 клеток (RD-модель). Сложность вычисления (табличного представления) булевых функций перехода в RD-модели вполне приемлема. Величину отклонения от полной изотропии аналитически не удалось получить. Однако вычислительные эксперименты по моделированию потока в трубе показали, что параболы Пуазейля в разных продольных разрезах трубы симметричны и неразличимы в пределах принятой в экспериментах точности. Отсюда сделан вывод, что модель применима там, где малое нарушение изотропии не играет роли, например при моделировании потока в пористой среде [25].

Преобразование результатов моделирования к привычным вещественным значениям скорости производится путём осреднения полученных значений по пространству и времени. Эта процедура сглаживает полученные зависимости, уменьшая или уничтожая так называемый «автоматный шум» (изломанный характер кривых, обусловленный дискретностью значений переменных). Для параллельных реализаций создан программный комплекс [26], работающий на кластере Межведомственного суперкомпьютерного центра РАН (г. Москва). На тестовых задачах эффективность параллельных реализаций при количестве процессоров $n < 500$ составила $> 90\,\%$, уменьшаясь до $70\,\%$ при увеличении числа процессоров до 1600.

2.3. Целочисленная КА-модель газового потока

Целочисленная LGA-модель потока (сначала названная автором мультичастичной) [21] предложена для смягчения негативных проявлений булевых представлений переменных: автоматного шума, неспособности моделировать турбулентные потоки, невозможности моделировать движение твёрдого тела в потоке и, следовательно, сжатия газа. Целочисленная модель отличается от булевой алфавитом состояний, который содержит не булевы, а целочисленные векторы: в двумерном случае $w_k = (u_1, \ldots, u_6)$,

где $u_i \in \mathbb{Z}$. Глобальный оператор, как и в булевой модели, является суперпозицией глобальных операторов сдвига (4) и столкновения (5). При этом для каждого состояния клетки правой части (5) функция переходов равна одному из всех возможных выбранных равновероятно значений $w' = (u'_1 \dots u'_6)$, удовлетворяющих (6). Целочисленность алфавита позволяет моделировать движение твёрдых объектов в газе, имитируя сжатие газа. Чтобы реализовать это свойство, разработаны специальные локальные операторы взаимодействия частиц с перемещающимся твёрдым телом. Моделирование движения поршня в трубе и вдвигающейся задвижки, а также формирования порошковой струи под действием взрыва показало качественное сходство с моделируемыми процессами [21]. Тестовые вычислительные эксперименты по моделированию потоков в трубе показали, что число Рейнольдса Re имеет значения $800 \div 1000$ и за препятствием хорошо видны отрывающиеся вихри.

2.4. Сравнение с непрерывными моделями

Модели КА-гидродинамики имеют аналоги в непрерывной математике. Поэтому имеет смысл провести сопоставление их свойств и возможностей с другими методами моделирования жидкостных потоков.

Трудности трёхмерной реализации и, главное, непривычность к булевому выражению физических величин привело к появлению более приближённой к традиционной математике модели, так называемой Lattice-Gas-Boltzmann (LBM) модели [27]. В ней дискретное состояние клеток заменено функцией распределения плотности, а функции столкновения — непрерывным оператором. По сути, модель получилась равной дискретизированной форме уравнения Больцмана. Идея построения вычислительного метода для моделирования гидро- и газодинамики в обход дифференциальных уравнений в частных производных реализована также в кинетически согласованных разностных схемах (КСРС) [28], где из кинетического описания среды на микроуровне выполняется непосредственный переход к разностным уравнениям. В отличие от LGA-моделей, которые находятся на стадии исследования, эти две модели хорошо развиты. Их распространение подтверждает целесообразность создания вычислительных алгоритмов моделирования пространственной динамики непосредственно из кинетических описаний. Это сокращает путь от представления моделируемого явления до его имитации на компьютере.

Действительно, для традиционных методов гидродинамики полный путь моделирования состоит из следующих этапов: 1) вывод дифференциального уравнения из законов сохранения; 2) дискретизация времени и пространства; 3) построение вычислительного метода; 4) программирование (чаще параллельное) и 5) компьютерная реализация (рис. 4). На каждом этапе происходит потеря точности. При этом на пер-

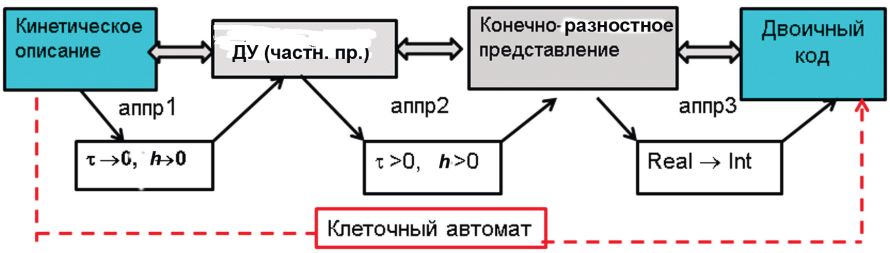

Рис. 4. Схема построения моделей пространственной динамики

вом этапе выполняется переход от дискретного микропредставления к непрерывному макропредставлению, а на втором — обратная операция дискретизации пространства

и времени. На третьем этапе надо сделать выбор между эффективно распараллеливаемыми явными вычислительными методами с ограничением Куранта и неявными методами с неэффективной суперкомпьютерной реализацией. Наконец, на последнем этапе компьютер аппроксимирует вещественные числа булевыми векторами заданной разрядности. Для моделей КСРС и LBM путь начинается с кинетического описания процесса, от которого, в каждом случае по-своему, совершается непосредственный переход к конечно-разностному представлению и явному методу программирования с более слабыми условиями Куранта. Наконец, LGA-модель, исходя из кинетического описания явления, выраженного булевыми функциями, совершает переход сразу к программе в двоичных кодах, что позволяет избежать преобразования чисел с плавающей точкой к двоичному коду. Это последнее исключает погрешности вычислений, вносимые округлением до заданной разрядности, и снимает проблему обеспечения вычислительной точности и вычислительной устойчивости. Так, в [29] показано, что при увеличении размеров области вычисления значений простых функций ($\sin(x)$) растёт размер ошибки округления. Поэтому при вычислениях на мощных суперкомпьютерах LGA-метод может оказаться предпочтительным.

3. Клеточно-автоматные модели, не имеющие непрерывных аналогов

3.1. Типичные компоненты сложных процессов

КА-модель процесса пространственного разделения смесей двух веществ на компоненты («разделения фаз») имеет булев алфавит $A = \{0, 1\}$, множество клеток $X = \{(i, j) : i, j = 0, \ldots, M\}$ — в виде декартовой решётки, контекст клетки (i, j) представлен квадратом с центром в этой клетке и содержит q клеток

$$\theta(i,j)[\{(u_0,(i,j))\},\{(u_1,\varphi_1(i,j)),\ldots,(u_q,\varphi_q(i,j))\}] = \{(v_0,(i,j))\}, \tag{7}$$

где

$$v_0 = \begin{cases} 1, \text{если} \quad \sum\limits_{k=0}^{q} u_k > (q+1)/2 \quad \text{или} \quad \sum\limits_{k=0}^{q} u_k < (q-1)/2, \\ 0, \text{если} \quad \sum\limits_{k=0}^{q} u_k < (q-1)/2 \quad \text{или} \quad \sum\limits_{k=0}^{q} u_k < (q+1)/2. \end{cases}$$

Режим работы КА-модели — синхронный. Из рис. 5, где показано четыре глобальных состояния из эволюции при исходном $\Omega(0)$, равном случайному равномерному распределению единиц в клеточном массиве с плотностью 0,5, видно, что процесс диссипативный (энтропия уменьшается).

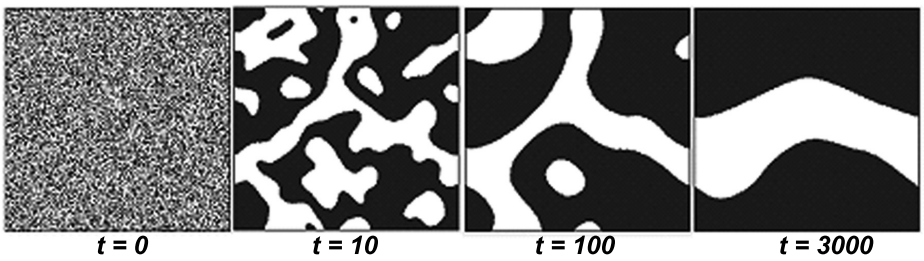

Рис. 5. Четыре глобальных состояния из эволюции КА «разделения фаз»

Ещё одна типичная простая КА-модель имитирует процесс формирования устойчивых образов [30]. В ней тоже используется булев алфавит и плоская декартова решётка.

Модель работает как в синхронном, так и в асинхронном режимах, хотя при этом эволюционирует совершенно по-разному. Локальный оператор имеет вид (7) с функцией перехода, заданной выражением

$$v_0 = \begin{cases} 1, \text{если} & \sum\limits_{k=0}^{q} u_k w_k > 0, \\ 0, \text{если} & \sum\limits_{k=0}^{q} u_k w_k \leqslant 0, \end{cases}$$

где w_k — вещественные числа, интерпретируемые как «веса» состояний контекстных клеток (если $w_k < 0$, то клетка — ингибитор, если $w_k > 0$, то активатор) (рис. 6).

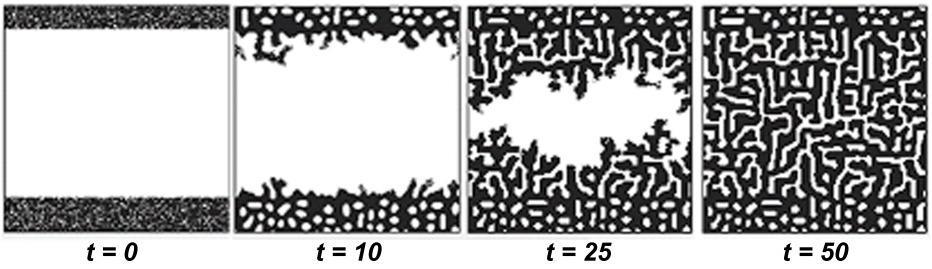

Рис. 6. Четыре глобальных состояния из эволюции асинхронной КА-модели процесса формирования устойчивого образа

3.2. КА-модель гетерогенной химической реакции на катализаторе

Репрезентативным примером КА-моделирования сложного процесса может служить гетерогенная реакция окисления монооксиси углерода $CO \to CO_2$ на поверхности катализатора [30, 31]. В связи с реконструкцией кристаллической решетки платины под воздействием адсорбированных молекул CO структура поверхности может быть гексагональной (hex) или кубической (1×1). Элементарные события (абсорбция, реакция, десорбция, перестройка структуры поверхности, диффузия, реакция и др.) описываются девятью химическими уравнениями. КА-модель $\aleph = \langle A, X, \Theta, (\alpha, \gamma) \rangle$ этого процесса имеет следующие характеристики: алфавит состояний

$$A = \{*_{1\times1},\ *_{\text{hex}},\ \text{CO}_{\text{ads}}^{1\times1},\ \text{CO}_{\text{ads}}^{\text{hex}},\ \text{O}_{\text{ads}}^{1\times1},\ \text{O}_{\text{ads}}^{\text{hex}}\}$$

содержит символы химических элементов, адсорбированных на поверхности двух типов; $X = \{(i,j) : i,j = 0,1,\ldots,M\}$, $\Theta(i,j) = \{\theta_1(i,j),\ldots,\theta_9(i,j)\}$; локальные операторы в Θ моделируют все элементарные действия процесса реакции. Глобальный оператор $\Theta(\Omega)$ работает в стохастическом режиме. Четыре глобальные состояния из эволюции процесса моделирования показаны на рис. 7. Моделирование выявило области параметров реакции, при которых процесс приходит к одному из возможных устойчивых состояний: стационарное однородное покрытие или автоколебания.

Реализация КА-модели на клеточном массиве размерами 8000×8000 выполнялась в блочно-синхронной версии на суперкомпьютере МВС-100К МСЦ РАН. Наибольшую эффективность распараллеливания (0,73 в сильном смысле) удалось получить при параллельном режиме на 128 ядрах.

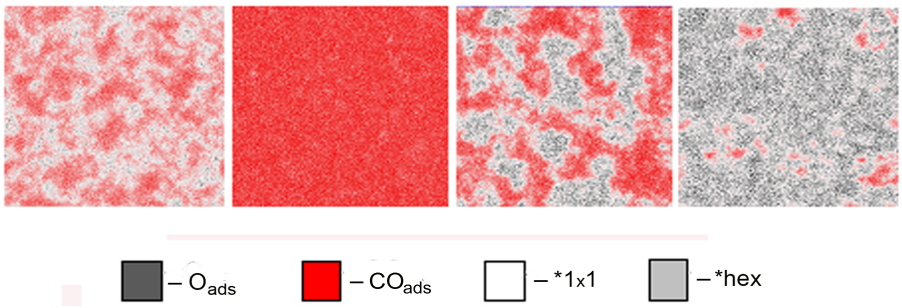

Рис. 7. Изменение состояний поверхности в ходе реакции $CO_{ads} \to O_{ads}$ на Pt_{100}. Слева направо: образование островков CO_{ads}, которые затем покрывают всю поверхность, на которой появляются островки O_{ads}, распространяющиеся по всей поверхности

4. Композиции глобальных операторов клеточных автоматов

4.1. Последовательная композиция

Ярким примером последовательной композиции (суперпозиции) глобальных операторов может служить КА-модель $\aleph_{soil} = \langle A, X, \varphi(\Theta)\rangle$ увлажнения почвы [32]. Алфавит состояний $A = \{w, s, g, h\}$ содержит символы, интерпретируемые как вода (w), твердая порода (s), газ (g) и разбухшее гидрофильное включение (h). Почва задана трёхмерным клеточным массивом $\Omega = \{(u, (i, j, k)) : u \in A,\ (i, j, k) \in X,\ i, j = 0, 1, \ldots, 700,\ k = 0, 1, \ldots, 1480\}$ (рис. 8). В начальном состоянии на поверхности почвы — слой воды. Моделируется постепенное проникновение этой воды в пористую структуру почвы (рис. 9) с одновременным испарением и с учётом влияния гидрофильных включений, которые запирают поры при разбухании и открывают их при высыхании.

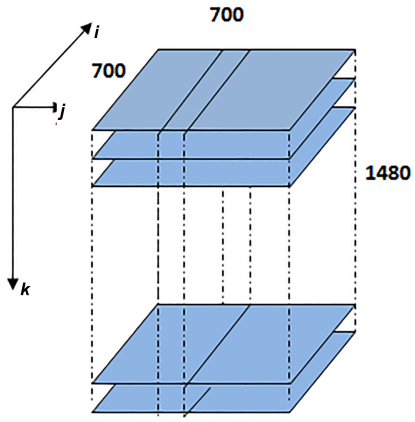

Рис. 8. Структура клеточного массива КА-модели увлажнения почвы

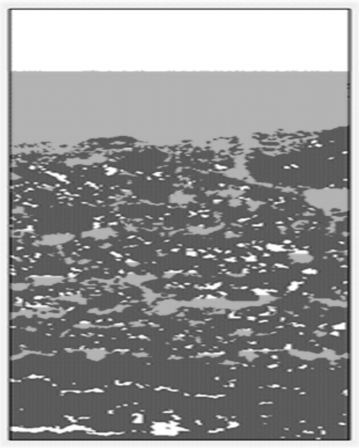

Рис. 9. Разрез фрагмента клеточного массива при $t = 10000$ итераций

Глобальный оператор построен как суперпозиция пяти простых глобальных операторов $\varphi(\Theta) = \Theta_G^{\kappa}(\Theta_E^{\kappa}(\Theta_S^{\beta}(\Theta_Q^{\beta}(\Theta_D^{\beta}(\Omega(0))))))$, где

— Θ_G^{κ} — оператор конвекции, содержащий локальный оператор

$$\theta_G(i, j, k)[\{(w, (i, j, k)), (g, (i, j, k + 1))\}] = \{(g, (i, j, k)), (w, (i, j, k + 1))\}, \qquad (8)$$

который моделирует движение частицы воды вниз под действием гравитации. Режим выполнения — асинхронный выбор клеток с вероятностью p_G, упорядоченный вдоль вертикальной оси k вниз;

— Θ_E^{κ} — оператор испарения, отличающийся от (8) взаимозаменой состояний w и g, значением вероятности p_E и направлением упорядочивания асинхронного выбора клеток (вдоль оси k вверх);
— Θ_S^{β} — оператор разбухания гидрофильной стенки, содержит локальный оператор

$$\begin{gathered}\theta_S(i,j,k)[\{(h,(i,j,k)),(w,\varphi_l(i,j,k)) : l=1,2,3,4\}] = \\ = \{(h,(i,j,k)),(h,\varphi_l(i,j,k)) : l=1,2,3,4\},\end{gathered} \tag{9}$$

где $\varphi_l(i,j,k) = (i+a_l, j+b_l, k)$ и a_l, b_l для каждого $l = 1,2,3,4$ определяются как первые два числа в тройке, выбранной с равной вероятностью из четырёх последних троек в соседстве $N(i,j,k) = \{(i,j,k),(0,1,k),(1,0,k),(0,-1,k),(-1,0,k)\}$;
— Θ_Q^{β} — оператор высыхания моделирует выделение воды из разбухшей стенки при высыхании, он отличается от (9) взаимозаменой состояний w и h;
— Θ_D^{β} — оператор диффузии моделирует выравнивание поверхности воды в порах, кавернах и на поверхности, для полного выравнивания поверхности выполняется 20 раз на каждой итерации.

Поскольку КА-модель ориентирована на параллельную реализацию, операторы диффузии и гидрофильного влияния применяются в блочно-синхронном режиме с девятью стадиями.

Важным свойством предложенной КА-модели является тот факт, что параллельный алгоритм вычислений не требует выполнения обменов в трёх измерениях. При декомпозиции клеточного массива на вертикальные параллелепипеды плоскостями (k,i) и (k,j) обмены данными между подмассивами происходят только в горизонтальных плоскостях. Всего необходимо $9 \cdot 22 = 198$ обменов на каждой итерации. Клеточный массив разрезался на $7 \cdot 7 = 49$ подмассивов, размещаемых на 49 ядрах кластера НКС-30 ССКЦ СО РАН (Intel Xeon E5540, 2,53 ГГц). Для выполнения 50 000 итераций потребовалось 10 часов при эффективности распараллеливания примерно 70 %.

5. Параллельная композиция глобальных операторов

Известно несколько КА-моделей с глобальным оператором в виде параллельной композиции $\Psi(\Theta_1(\Omega),\ldots,\Theta_L(\Omega))$ [19, 33]. Большинство известных КА-моделей [30, 34] используют два глобальных оператора. Один из них моделирует исследуемое явление, а второй является контекстным, так как моделирует влияющие на него изменяющиеся внешние условия.

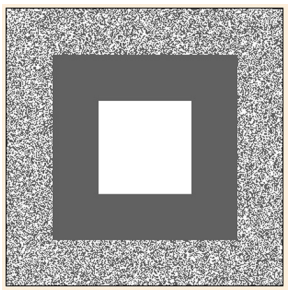

Рис. 10. Начальное состояние $\Omega_v(0)$

Типичным примером служит КА-модель процесса формирования узора на поверхности остывающей металлической пластины $\aleph = \langle A, X, \Psi(\Theta)\rangle$, где $\Theta = \{\Theta_u^{\alpha}, \Theta_v^{\alpha}\}$ и соответственно $X = X_u \cup X_v$ с клетками $(i,j)_u$ в X_u и $(i,j)_v$ в X_v и $\Omega = \Omega_u \cup \Omega_v$. Оба оператора простые, функционируют асинхронно, имеют булев алфавит состояний клеток. Оператор Θ_u^{α} имитирует распространение тепла по пластине от горячей области, где $u = 1$, к холодному квадрату в центре ($u = 0$) и к тёплым краям пластины (рис. 10) путём применения локальных операторов диффузии $\theta_l(i,j)$ (3). Эволюция этой модели имитирует изменение распределения температуры пластины.

Простой глобальный оператор Θ_v^{α} выполняет основную функцию формирования устойчивого образа (узора) путём применения к $\Omega_v(t)$ локального оператора тоталистического типа [8] $\theta_{\mathrm{tot}}(i,j)$, у которого контекст $C((i,j)_v)$ состоит из двух частей:

$C'((i,j)_v) = \{(v_{i+a,j+b},(i+a,j+b)_v) : a,b = -r,-r+1,\ldots,0,1,\ldots,r+1\} \setminus \{(i,j)\}$, $C((i,j)_u) = \{(u_{i+a,j+b},(i+a,j+b)_u) : a,b = -r,-r+1,\ldots,0,1,\ldots,r+1\}$, $r \ll |X|$ — радиус соседства $\theta_{\text{tot}}(i,j)[\{(v_0,(i,j)_v)\}, C'((i,j)_v) \cup C((i,j)_u)] = \{(v'_0,(i,j)_v)\}$, где v'_0 вычисляется функцией переходов как

$$v'_0 = \begin{cases} 1, & \text{если } \sum\limits_{a,b=-r}^{r+1} v_{i+a,j+b} \cdot w(i+a,j+b) \geqslant 0, \\ 0, & \text{если } \sum\limits_{a,b=-r}^{r+1} v_{i+a,j+b} \cdot w(i+a,j+b) < 0. \end{cases}$$

Здесь $w(i+a,j+b) = -0{,}2\, u_{i+a,j+b} \cdot (-1)^{(a+b)}$.

В начальном массиве $\Omega_v(0)$ значения $u = 1$ равномерно распределены с плотностью 0,5.

На рис. 11 показаны три глобальных состояния $\Omega_v(t)$. Моделирование выполнялось на двух ядрах процессора Intel Core i-4770K. Параллельная работа глобальных операторов обеспечивалась с помощью библиотеки OpenMP. Устойчивое состояние на клеточном массиве размером 2×400 достигалось за 2,5 мин.

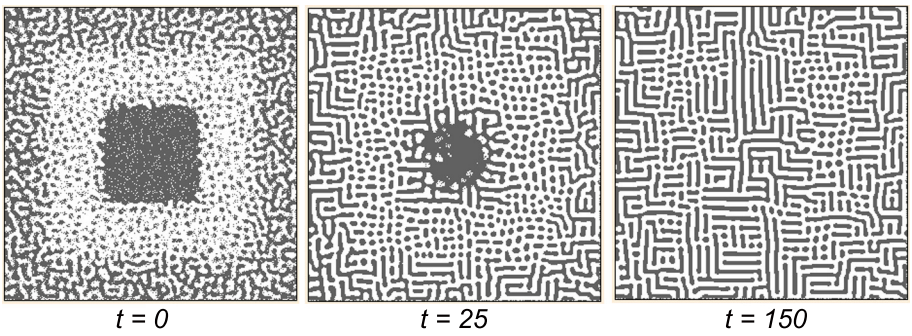

Рис. 11. Три глобальных состояния $\Omega_v(t)$, полученных при моделировании процесса формировния узора на остывающей металлической пластине

Заключение

Клеточно-автоматные модели пространственной динамики — это одно из направлений в математическом и компьютерном моделировании, которое согласуется с современными тенденциями параллелизации архитектуры вычислительных систем. Основные свойства КА-моделей — локальность взаимодействий вычисляющих операторов и отказ от вещественных переменных — избавляют от острых проблем вычислительной математики: проблемы распараллеливания, проблемы потери точности и, наконец, проблемы сложности программирования. Более того, вычислительные свойства КА-моделей — способность моделировать нелинейные явления (самоорганизация, автоволны, образование фрактальных структур) — согласуются с насущной потребностью развивать методы моделирования химических и биологических процессов. Автор сочтёт, что его цель достигнута, если эта работа убедит кого-то из специалистов в необходимости исследовать и развивать КА-моделирование.

ЛИТЕРАТУРА

1. *Фон Нейман Дж.* Теория самовоспроизводящихся автоматов. М.: Мир, 1971. 384 с.
2. *Wolfram S.* Computation theory of cellular automata // Commun. Math. Phys. 1984. V. 96. P. 15–57.
3. *Sandefur J. T.* Discrete Dynamical Systems: Theory and Applications. Oxford: University Press, 1990. 445 p.
4. *Donienunzio A., Formenti E., and Kurka P.* Cellular automata dynamical systems // Handbook of Natural Computing. V. 1. / eds. G. Rozenberg, T. Bäck, J. N. Kok. Berlin: Springer, 2012. P. 24–76.
5. *Wolfram S.* Statistical mechanics of cellular automata // Rev. Mod. Phys. 1983. V. 35. P. 601–643.
6. *Toffolli T. and Margolus N.* Cellular Automata Machines: A New Environment for Modeling. USA: MIT-Press, 1987. 260 p.
7. *Frish U., Hasslacher B., and Pomeau Y.* Lattice-gas automata for Navier — Stokes equation // Phys. Rev. Let. 1986. V. 56. P. 1505–1508.
8. *Wolfram S.* A New Kind of Science. Champaign, Ill. USA: Wolfram Media Inc., 2002. 1197 p.
9. *Weimar J.* Cellular automata for reaction-diffusion systems // Parallel Computing. 1997. V. 23. No. 11. P. 1699–1715.
10. *Ванаг В. К.* Диссипативные структуры в реакционно-диффузионных системах. Ижевск: Ин-т компьют. исслед., 2008. 300 с.
11. *Madore B. and Freedman W.* Computer simulation of the Belousov — Zhabotinski reaction // Science. 1983. V. 222. P. 615–618.
12. *Park J. K., Steiglitz K., and Thurston W. P.* Soliton-like behavior in automata // Physica D. 1986. V. 19. P. 423–432.
13. *Achasova S., Bandman O., Markova V., and Piskunov S.* Parallel Substitution Algorithm. Theory and Application. Singapore: World Scientific, 1994. 198 p.
14. *Бандман О. Л.* Клеточно-автоматные модели пространственной динамики // Системная информатика. 2006. Вып. 10. С. 59–111.
15. *Ачасова С. М., Бандман О. Л.* Корректность параллельных вычислительных процессов. Новосибирск: Наука, 1990. 256 с.
16. *Bandman O.* Parallel simulation of asynchronous cellular automata evolution // ACRI-2006. LNCS. 2006. V. 4173. P. 41–47.
17. *Sharifulina A. and Elokhin V.* Simulation of heterogeneous catalytic reaction by asynchronous cellular automata on multicomputer // PaCT-2011. LNCS. 2011. V. 6873. P. 210–216.
18. *Шарифулина А. Е.* Параллельная реализация каталитической реакции ($CO + O_2 \to CO_2$) с помощью асинхронного клеточного автомата // Труды ПаВТ-2012. Челябинск: Изд-во ЮрГУ, 2012. С. 325–335.
19. *Bandman O.* Cellular automata composition techniques for spatial dynamics simulation // Simulating Complex Systems by Cellular Automata / eds. A. G. Hoekstra, J. Kroc, P. M. A. Sloot. Berlin: Springer, 2010. P. 81–116.
20. *Bandman O.* Using multi core computers for implementing cellular automata systems // PaCT-2011. LNCS. 2011. V. 6873. P. 45–58.
21. *Медведев Ю. Г.* Многочастичная клеточно-автоматная модель потока жидкости FHP-MP // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2009. № 1(6). С. 33–40.
22. *Медведев Ю. Г.* Моделирование трехмерных потоков клеточными автоматами // Вестник Томского государственного университета. 2002. Приложение. № 1 (II). С. 236–240.

23. *Медведев Ю. Г.* Многочастичные клеточно-автоматные модели диффузионного процесса // Новые инф. технологии в исслед. сложных структур. Тез. докл. Томск: Изд-во НТЛ, 2010. С. 21–22.
24. *Bandman O.* Cellular automata diffusion models for multicomputer implementation // Bull. Nov. Comp. Center. Ser. Comp. Sci. 2014. V. 36. P. 21–31.
25. *Медведев Ю. Г.* Применение клеточно-автоматной модели потока вязкой жидкости в исследовании трёхмерных пористых сред // Автометрия. 2006. Т. 42. № 3. С. 21–31.
26. *Медведев Ю. Г.* Программный комплекс клеточно-автоматного моделирования газопорошковых потоков // Труды ПаВТ-2012. Челябинск: Изд-во ЮрГУ, 2012. С. 732.
27. *Wolf-Gladrow D.* Lattice-Gas Cellular Automata and Lattice Boltzmann Models. Berlin: Springer, 2000. 310 p.
28. *Четверушкин Б. Н.* Кинетически-согласованные схемы в газовой динамике: новая модель вязкого газа, алгоритмы, параллельная реализация, приложения. М.: Изд-во МГУ, 1999. 232 с.
29. Numerical Algorithms and Round-off Errors. `http://www.sml.ee.upatras.gr/uploadedfiles/roundofferror.pdf`.
30. *Kireeva A.* Two-layer CA model for simulating catalytic reaction at dynamically varying temperature // ACRI-2014. LNCS. 2014. V. 8751. P. 166–175.
31. *Бандман О. Л., Киреева А. Е.* Стохастическое клеточно-автоматное моделирование колебаний и автоволн в реакционно-диффузионных системах // Сиб. ЖВМ. 2015. Т. 18. № 3. С. 255–274.
32. *Bandman O.* Cellular automata model of fluid permeation through porous material // PaCT-2013. LNCS. 2013. V. 7979. P. 295–316.
33. *Bandman O.* Parallel composition of asynchronous cellular automata simulating reaction diffusion processes // ACRI-2010. LNCS. 2010. V. 6350. P. 395–398.
34. *Витвицкий А.* Клеточные автоматы с динамической структурой для моделирования роста биологических тканей // Сиб. ЖВМ. 2014. Т. 17. № 4. С. 315–325.

REFERENCES

1. *Von Neumann J.* Teoriya samovosproizvodyashchikhsya avtomatov [The Theory of Self-Reproducing Automata]. Moscow, Mir Publ., 1971. 384 p. (in Russian)
2. *Wolfram S.* Computation theory of cellular automata. Commun. Math. Phys., 1984, vol. 96, pp. 15–57.
3. *Sandefur J. T.* Discrete Dynamical Systems: Theory and Applications. Oxford: University Press, 1990. 445 p.
4. *Donienunzio A., Formenti E., and Kurka P.* Cellular automata dynamical systems. Handbook of Natural Computing. V. 1. / eds. G. Rozenberg, T. Bäck, J. N. Kok. Berlin: Springer, 2012, pp. 24–76.
5. *Wolfram S.* Statistical mechanics of cellular automata. Rev. Mod. Phys., 1983, vol. 35, pp. 601–643.
6. *Toffolli T. and Margolus N.* Cellular Automata Machines: A New Environment for Modeling. USA: MIT-Press, 1987. 260 p.
7. *Frish U., Hasslacher B., and Pomeau Y.* Lattice-gas automata for Navier — Stokes equation. Phys. Rev. Let., 1986, vol. 56, pp. 1505–1508.
8. *Wolfram S.* A New Kind of Science. Champaign, Ill. USA: Wolfram Media Inc., 2002. 1197 p.
9. *Weimar J.* Cellular automata for reaction-diffusion systems. Parallel Computing, 1997, vol. 23, no. 11, pp. 1699–1715.

10. *Vanag V. K.* Dissipativnye struktury v reaktsionno-diffuzionnykh sistemakh [Dissipative Structures in Reaction-Diffusion Systems]. Izhevsk, Institute of Computer Science Publ., 2008. 300 p. (in Russian)
11. *Madore B. and Freedman W.* Computer simulation of the Belousov — Zhabotinski reaction. Science, 1983, vol. 222, pp. 615–618.
12. *Park J. K., Steiglitz K., and Thurston W. P.* Soliton-like behavior in automata. Physica D., 1986, vol. 19, pp. 423–432.
13. *Achasova S., Bandman O., Markova V., and Piskunov S.* Parallel Substitution Algorithm. Theory and Application. Singapore: World Scientific, 1994. 198 p.
14. *Bandman O. L.* Kletochno-avtomatnye modeli prostranstvennoy dinamiki [Cellular automata models of spatial dynamics]. System Informatics, 2006, no. 10, pp. 59–111. (in Russian)
15. *Achasova S. M., Bandman O. L.* Korrektnost' parallel'nykh vychislitel'nykh protsessov [The Correctness of Parallel Computing Processes]. Novosibirsk, Nauka Publ., 1990. 256 p. (in Russian)
16. *Bandman O.* Parallel simulation of asynchronous cellular automata evolution. ACRI-2006, LNCS, 2006, vol. 4173, pp. 41–47.
17. *Sharifulina A. and Elokhin V.* Simulation of heterogeneous catalytic reaction by asynchronous cellular automata on multicomputer. PaCT-2011, LNCS, 2011, vol. 6873, pp. 210–216.
18. *Sharifulina A. E.* Parallel'naya realizatsiya kataliticheskoy reaktsii ($SO + O_2 \rightarrow SO_2$) s pomoshch'yu asinkhronnogo kletochnogo avtomata [Parallel implementation of the catalytic reaction ($SO + O_2 \rightarrow SO_2$) using an asynchronous cellular automaton]. Proc. PaVT-2012, Chelyabinsk, SUSU Publ., 2012, pp. 325–335. (in Russian)
19. *Bandman O.* Cellular automata composition techniques for spatial dynamics simulation. Simulating Complex Systems by Cellular Automata / eds. A. G. Hoekstra, J. Kroc, P. M. A. Sloot. Berlin, Springer, 2010, pp. 81–116.
20. *Bandman O.* Using multi core computers for implementing cellular automata systems. PaCT-2011, LNCS, 2011, vol. 6873, pp. 45–58.
21. *Medvedev Yu. G.* Mnogochastichnaya kletochno-avtomatnaya model' potoka zhidkosti FHP-MP [An extension of the cellular-automaton FHP-I flow model to the FHP-MP multi-particle model]. Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika, 2009, no. 1(6), pp. 33–40. (in Russian)
22. *Medvedev Yu. G.* Modelirovanie trekhmernykh potokov kletochnymi avtomatami [The 3D fluid flow simulation]. Vestnik Tomskogo gosudarstvennogo universiteta. Prilozhenie, 2002, no. 1(II), pp. 236–240. (in Russian)
23. *Medvedev Yu. G.* Mnogochastichnye kletochno-avtomatnye modeli diffuzionnogo protsessa [Many-cellular automata models of diffusion process]. Novye inf. tekhnologii v issled. slozhnykh struktur. Tez. dokl. Tomsk, NTL Publ., 2010, pp. 21–22. (in Russian)
24. *Bandman O.* Cellular automata diffusion models for multicomputer implementation. Bull. Nov. Comp. Center. Ser. Comp. Sci., 2014, vol. 36, pp. 21–31.
25. *Medvedev Yu. G.* Primenenie kletochno-avtomatnoy modeli potoka vyazkoy zhidkosti v issledovanii trekhmernykh poristykh sred [Applying the cellular automaton model of a viscous fluid flow in investigating 3D porous media. Avtometriya, 2006, vol. 42, no. 3, pp. 21–31. (in Russian)
26. *Medvedev Yu. G.* Programmnyy kompleks kletochno-avtomatnogo modelirovaniya gazoporoshkovykh potokov [The software package for cellular automata modeling of gas-powder flow]. Proc. PaVT-2012. Chelyabinsk, SUSU Publ., 2012, p. 732. (in Russian)
27. *Wolf-Gladrow D.* Lattice-Gas Cellular Automata and Lattice Boltzmann Models. Berlin, Springer, 2000. 310 p.

28. *Chetverushkin B. N.* Kineticheski-soglasovannye skhemy v gazovoy dinamike: novaya model' vyazkogo gaza, algoritmy, parallel'naya realizatsiya, prilozheniya [Kinetically consistent schemes in gas dynamics: the new model of a viscous gas, algorithms, parallel implementation, application]. Moscow, MSU Publ., 1999. 232 p. (in Russian)
29. Numerical Algorithms and Round-off Errors. `http://www.sml.ee.upatras.gr/uploadedfiles/roundofferror.pdf`.
30. *Kireeva A.* Two-layer CA model for simulating catalytic reaction at dynamically varying temperature. ACRI-2014, LNCS, 2014, vol. 8751, pp. 166–175.
31. *Bandman O. L. and Kireeva A. E.* Stochastic cellular automata simulation of oscillations and autowaves in reaction-diffusion systems. Numerical Analysis and Applications, 2015, vol. 8, no. 3, pp. 208–222.
32. *Bandman O.* Cellular automata model of fluid permeation through porous material. PaCT-2013, LNCS, 2013, vol. 7979, pp. 295–316.
33. *Bandman O.* Parallel composition of asynchronous cellular automata simulating reaction diffusion processes. ACRI-2010, LNCS, 2010, vol. 6350, pp. 395–398.
34. *Vitvitsky A. A.* Cellular automata with dynamic structure to simulate the growth of biological tissues. Numerical Analysis and Applications, 2014, vol. 7, no. 4, pp. 263–273.

СВЕДЕНИЯ ОБ АВТОРАХ

АВЕЗОВА Яна Эдуардовна — аспирантка Национального исследовательского ядерного университета «МИФИ», г. Москва. E-mail: **avezovayana@gmail.com**

АГИБАЛОВ Геннадий Петрович — доктор технических наук, профессор, профессор кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: **agibalov@isc.tsu.ru**

БАНДМАН Ольга Леонидовна — доктор технических наук, профессор, главный научный сотрудник Института вычислительной математики и математической геофизики СО РАН, г. Новосибирск. E-mail: **bandman@ssd.sscc.ru**

ЖУКОВ Кирилл Дмитриевич — Лаборатория ТВП, г. Москва.
E-mail: **zhukov_kirill@bk.ru**

КАБЕНЮК Михаил Иванович — кандидат физико-математических наук, доцент Кемеровского государственного университета, г. Кемерово.
E-mail: **kabenyuk@kemsu.ru**

КОСОЛАПОВ Юрий Владимирович — кандидат технических наук, доцент кафедры алгебры и дискретной математики Института математики, механики и компьютерных наук им. И. И. Воровича, Южный федеральный университет, г. Ростов-на-Дону. E-mail: **itaim@mail.ru**

МЕЖЕННАЯ Наталья Михайловна — кандидат физико-математических наук, доцент кафедры прикладной математики Московского государственного технического университета имени Н. Э. Баумана, г. Москва.
E-mail: **natalia.mezhennaya@gmail.com**

ПАНКРАТОВА Ирина Анатольевна — кандидат физико-математических наук, доцент, доцент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.
E-mail: **pank@isc.tsu.ru**

ПЕТУХОВ Андрей Александрович — младший научный сотрудник лаборатории безопасности информационных систем факультета ВМК Московского государственного университета имени М. В. Ломоносова, г. Москва.
E-mail: **petand@seclab.cs.msu.su**

РАЗДОБАРОВ Александр Вячеславович — аспирант лаборатории безопасности информационных систем факультета ВМК Московского государственного университета имени М. В. Ломоносова, г. Москва. E-mail: **korse@seclab.cs.msu.su**

СИГАЛОВ Даниил Алексеевич — студент лаборатории безопасности информационных систем факультета ВМК Московского государственного университета имени М. В. Ломоносова, г. Москва. E-mail: **asterite@seclab.cs.msu.su**

ТУРЧЕНКО Олег Юрьевич — студент Института математики, механики и компьютерных наук им. И. И. Воровича, Южный федеральный университет, г. Ростов-на-Дону. E-mail: **olegmmcs@gmail.com**

ФОМИЧЕВ Владимир Михайлович — доктор физико-математических наук, профессор, профессор Финансового университета при Правительстве Российской Федерации, профессор Национального исследовательского ядерного университета «МИФИ», ведущий научный сотрудник ФИЦ ИУ РАН, научный консультант Службы сертификации ООО «Код Безопасности», г. Москва. E-mail: **fomichev.2016@yandex.ru**

ЧЕРЕМУШКИН Александр Васильевич — доктор физико-математических наук, ФГУП «НИИ «Квант», г. Москва. E-mail: **avc238@mail.ru**

Журнал «Прикладная дискретная математика» включен в перечень ВАК рецензируемых российских журналов, в которых должны быть опубликованы основные результаты диссертаций, представляемых на соискание учёной степени кандидата и доктора наук, в базы данных Web of Science (Russian Science Citation Index — RSCI) и Scopus, а также в перечень журналов, рекомендованных ФУМО ВО ИБ в качестве учебной литературы по специальности «Компьютерная безопасность».

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 в журнал.

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*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и её приложениям*