ISSN 1684-8853 (print); ISSN 2541-8610 (online)

ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

НАУЧНЫЙ ЖУРНАЛ

1(116)/2022

doi:10.31799/1684-8853-2022-1 ISSN 1684-8853 (print); ISSN 2541-8610 (online)

SCIENTIFIC JOURNAL

1(116)/2022

PEER REVIEWED JOURNAL

INFORMATSIONNO-UPRAVLIAIUSHCHIE SISTEMY (INFORMATION AND CONTROL SYSTEMS)

Contact information
The Editorial and Publishing Center, SUAI
67A, B. Morskaia, 190000, Saint Petersburg, Russia
Website: http://i-us.ru/en, e-mail: ius.spb@gmail.com
Tel.: +7 - 812 494 70 02

The Journal was registered in the Ministry of Press, Broadcasting and Mass Media of the Russian Federation.
Registration Certificate JD № 77-12412 from April, 19, 2002.
Re-registration in the Federal Service for Supervision in the Sphere of Telecom, Information Technologies and Mass Communications (ROSKOMNADZOR) due to change of the founder A. Vostrikov,
PI № FS 77-82226 from November, 23, 2021

Submitted for publication 11.01.2022. Passed for printing 24.02.2022. Signed for printing: 28.02.2022.
Format 60×84$_{1/8}$.Phototype SchoolBookC. Digital printing.

Layout original is made at the Editorial and Publishing Center, SUAI.
67A, B. Morskaia, 190000, Saint Petersburg, Russia
Printed from slides at the Editorial and Publishing Center, SUAI.
67A, B. Morskaia, 190000, Saint Petersburg, Russia

The journal is indexed in Scopus.
Free distribution.

doi:10.31799/1684-8853-2022-1 ISSN 1684-8853 (print); ISSN 2541-8610 (online)

НАУЧНЫЙ ЖУРНАЛ

1(116)/2022

РЕЦЕНЗИРУЕМОЕ ИЗДАНИЕ

ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

Учредитель
А. А. Востриков
Издатель
Санкт-Петербургский государственный университет аэрокосмического приборостроения
Главный редактор
Е. А. Крук,
д-р техн. наук, проф., Москва, РФ
Ответственный секретарь
О. В. Муравцова

Редактор: А. Г. Ларионова
Корректор: Т. В. Звертановская
Дизайн: М. Л. Черненко, Ю. В. Умницына
Компьютерная верстка: Ю. В. Умницына

Адрес редакции: 190000, г. Санкт-Петербург,
ул. Б. Морская, д. 67, лит. А, ГУАП, РИЦ
Тел.: (812) 494-70-02, эл. адрес: ius.spb@gmail.com,
сайт: http://i-us.ru

Журнал входит в БД SCOPUS и в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук.

Сдано в набор 11.01.2022. Подписано в печать 24.02.2022. Дата выхода в свет: 28.02.2022.
Формат 60×84 1/8. Гарнитура SchoolBookC. Печать цифровая.
Усл. печ. л. 8,3. Уч.-изд. л. 11,4. Тираж 1000 экз (1-й завод 50 экз.). Заказ № 62.

Оригинал-макет изготовлен в редакционно-издательском центре ГУАП.
190000, г. Санкт-Петербург, ул. Б. Морская, д. 67, лит. А.

Отпечатано в редакционно-издательском центре ГУАП.
190000, г. Санкт-Петербург, ул. Б. Морская, д. 67, лит. А.

Распространяется бесплатно.

Журнал зарегистрирован в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-12412 от 19 апреля 2002 г.
Перерегистрирован в Роскомнадзоре.
Свидетельство о регистрации ПИ № ФС77-82226 от 23 ноября 2021 г.

УДК 519.614
doi:10.31799/1684-8853-2022-1-2-7

Научные статьи
Articles

Критские матрицы Одина и Тени, сопровождающие простые числа и их степени

Н. А. Балонин[a]***,*** *доктор техн. наук, профессор, orcid.org/0000-0001-7338-4920, korbendfs@mail.ru*
М. Б. Сергеев[a]***,*** *доктор техн. наук, профессор, orcid.org/0000-0002-3845-9277*
[a]*Санкт-Петербургский государственный университет аэрокосмического приборостроения, Б. Морская ул., 67, Санкт-Петербург, 190000, РФ*

Введение: *критские матрицы — ортогональные матрицы, состоящие из элементов 1 и $-b$ (вещественное число), представляют собой идеальный объект для наглядного приложения конечномерной математики. К ним относятся, в частности, матрицы Адамара и, при расширении числа элементов, конференц-матрицы. Наиболее удобный аппарат исследования состоит в привлечении теории полей и мультипликативных групп Галуа, что особенно актуально для новых типов критских матриц.* ***Цель:*** *изучить симметрии критских матриц и исследовать два выделенных симметриями новых типа матриц нечетного и четного порядков соответственно, существенно отличающихся от ранее известных матриц Мерсенна, Эйлера и Ферма.* ***Результаты:*** *приведены формулы для значений элементов и описаны симметрии новых критских матриц: бициклов Одина (с каймой) порядков $4t-1$ и $4t-3$ и матриц Тени порядков $4t-2$ и $4t-4$. Для нечетных порядков матриц, равных простым числам и степеням простых чисел характеристических размеров, доказано существование симметрий особых типов этих матриц, двоякосимметричных, состоящих из кососимметричного (по знакам элементов) и симметричного циклических блоков. Показано, что ранее выделенные критские матрицы Мерсенна порядков $4t-1$ и Эйлера порядков $4t-2$ являются их частным случаем, существующим при отсутствии симметрии для всех выделенных порядков без исключения.* ***Практическая значимость:*** *ортогональные последовательности и методы их эффективного нахождения теорией конечных полей и групп имеют непосредственное практическое значение для задач помехоустойчивого кодирования, сжатия и маскирования видеоинформации.*

Ключевые слова *— матрицы Адамара, матрицы Белевича, критские матрицы, конечные поля, симметрии матриц.*

Для цитирования: Балонин Н. А., Сергеев М. Б. Критские матрицы Одина и Тени, сопровождающие простые числа и их степени. *Информационно-управляющие системы*, 2022, № 1, с. 2–7. doi:10.31799/1684-8853-2022-1-2-7
For citation: Balonin N. A., Sergeev M. B. Odin and Shadow Cretan matrices accompanying primes and their powers. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 2–7 (In Russian). doi:10.31799/1684-8853-2022-1-2-7

Введение

Впервые наследование порядками n матриц со значениями элементов (уровнями) 1 и –1 и ортогональных в смысле $\mathbf{H}^{\mathrm{T}}\mathbf{H} = n\mathbf{I}$, где $\mathbf{I}$ — единичная матрица, значений числовых последовательностей заметил еще основоположник теории матриц Дж. Сильвестр [1, 2]. Адамар [3] дополнил это наблюдение вложением ряда степеней простого числа 2^k в более широкую числовую последовательность вида $n = 4t$, где t — натуральное число, самостоятельно найдя матрицы порядков 12 и 20.

С тех пор в теории матриц Адамара достигнуты большие успехи [4, 5]. Вычисление матрицы размера 428 [6] подняло планку нижнего неизвестного пока порядка матриц Адамара до 668. Обобщающие их критские матрицы [7, 8] были введены в рамках композиционного построения теории экстремальных по детерминанту матриц [9], сопровождающих другие известные в теории числовые последовательности.

Изучению свойств симметрии критских матриц, а также двух новых видов симметричных матриц нечетного и четного порядков, существенно отличающихся от ранее известных матриц, посвящена настоящая работа.

Критские матрицы

Критские матрицы $\mathbf{K}$ во многом похожи на матрицы Адамара [10] (даже больше, чем взвешенные матрицы [11] с тремя уровнями на порядках, кратных двум). Это столь же малоуровневые матрицы с элементами 1 и $-b$, не превосходящими по модулю единицы, для которых справедливо $\mathbf{K}^{\mathrm{T}}\mathbf{K} = \omega\mathbf{I}$, где $\omega \le 1$ — некоторый весовой коэффициент [7, 11]. Число уровней в критских матрицах расширяемо, например, элементом d на диагонали. Фиксировать семейства критских матриц можно, как у матриц Адамара, указывая характер экстремума или предложением формулы для уровней $b = b(n)$ и $d = d(n)$.

Семейство критских матриц шире семейства матриц Адамара и включает его. Например, критские матрицы при $b = 1$ — это классические матрицы Адамара $\mathbf{H}$, при $b = 1$ и $d = 0$ (на диагонали) — это матрицы Белевича, для которых справедливо иное условие ортогональности $\mathbf{C}^{\mathrm{T}}\mathbf{C} = (n-1)\mathbf{I}$. Уровни и условие ортогональности вполне однозначно идентифицируют матрицы, однако сами по себе формулы для уровней можно получить не априори, а апостериори, анализируя частные экстремумы детерминантов на серии за-

даваемых анализируемыми последовательностями (прямо или косвенно) порядков [9].

Матрицы Адамара сопровождают порядки $n = 2^k$ и четные числа $4t$, в которые степени двойки вложены (гипотеза Адамара), а матрицы Белевича (взвешенные матрицы, конференц-матрицы [11]) варьированием их диагонального уровня в 0 охватывают дополнительные порядки, равные числам вида $4t - 2$, разложимым на сумму двух квадратов. Последнее утверждение тоже является гипотезой [1], поскольку вид экстремальных матриц усложняется с ростом порядка настолько, что первые неразрешенные теорией случаи охватывают числа 66 и 86 (а не 668, как у матриц Адамара).

Матрицы Мерсенна **M** [12], относящиеся к критским, сопровождают порядки, равные числам последовательности Мерсенна $n = 2^k - 1$ и нечетным числам вида $4t - 1$, в которые отмеченная последовательность вложена, и отличаются иррациональным уровнем $b = \frac{t}{t + \sqrt{t}}$.

На настоящее время не известно, ограничено ли множество простых чисел Мерсенна или их количество конечно (как и у чисел Ферма). Следующая задача, которая нас интересует, состоит в выделении критских матриц порядков, сопровождающих простые числа и степени простых чисел вида $4t - 1$ и $4t - 3$.

Случай $4t - 1$ наиболее прост тем, что существование конечного поля $GF(n)$ гарантирует наличие кососимметичных (по знакам элементов, не уровням) матриц Мерсенна [13]. Здесь существенно то обстоятельство, что, как и у матриц Адамара, диагональ этих матриц варьируется по уровню от 0 до 1 без потери ортогональности при условии изменения уровня b. Собственно, матрицы Белевича **C** трансформируются в матрицы Адамара $\mathbf{C} = \mathbf{H} + \mathbf{I}$ ровно по такому же принципу — предварительно нужно добиться кососимметрии **C**, однако взаимные переходы в этом частном случае не изменяют отрицательный уровень.

В общем случае фиксация диагонали уменьшает количество возможных порядков с $4t - 1$ или $4t - 3$ до желаемого множества степеней простых чисел, причем отличаться матрицы будут симметриями. Такие критские матрицы ранее не рассматривались, поэтому мы их опишем максимально общо, не апеллируя к бициклической форме.

Матрицы Одина и матрицы Тени

Рассматриваемые матрицы могут получаться из хаотических матриц (оптимизация детерминанта не накладывает требований поддерживать структуру) или из матриц Адамара или Белевича, с основами (core) которых они тесно связаны при доказательствах теорем существования [12].

Определение 1. Матрица Одина порядка $4t - 1$, являющегося простым числом или его степенью, — это критская матрица с уровнями 1, $-b$, $d = 0$ (на диагонали), где $b = \frac{\upsilon - 1}{\upsilon + \sqrt{2\upsilon - 1}}$, $\upsilon = (n - 1)/2$ — половинный размер матрицы, за исключением ее каймы d.

Определение 2. Матрица Одина порядка $4t - 3$, являющегося простым числом или его степенью, — это критская матрица с уровнями 1, $-b$, $d = \frac{1}{1 + \sqrt{n}}$ (на диагонали), где $b = 1 - 2d$.

Инвариантом матрицы Одина является равное число внедиагональных уровней. Такая структура позволяет с легкостью выделить в качестве первой строки и столбца кайму из элементов векторов **e** и $-b\mathbf{e}$, где **e** — вектор из 1 длины υ. Структуры обеих матриц описываются кососимметричной (по знакам элементов) и симметричной формами соответственно:

$$\mathbf{O}_{4t-1} = \begin{pmatrix} d & \mathbf{e} & -b\mathbf{e} \\ -b\mathbf{e} & \mathbf{A} & \mathbf{B} \\ \mathbf{e} & [-\mathbf{B}^{\mathrm{T}}] & \mathbf{D}^{\mathrm{T}} \end{pmatrix};$$

$$\mathbf{O}_{4t-3} = \begin{pmatrix} d & -b\mathbf{e} & \mathbf{e} \\ -b\mathbf{e} & \mathbf{A} & \mathbf{B} \\ \mathbf{e} & \mathbf{B}^{\mathrm{T}} & [-\mathbf{D}^{\mathrm{T}}] \end{pmatrix}.$$

Здесь операция, обозначенная как $[\cdot]$, означает характерную для трехуровневых матриц замену всех положительных элементов транспонированной матрицы на 1 и всех отрицательных на $-b$. Добавление каймы из 1 и -1 в строке и столбце (с учетом симметрий) порождает матрицу Белевича **C** с уровнями $-b = -1$, $d = 0$. Отделение каймы у матриц Одина ведет к критским матрицам Тени **T** (shadow matrices) вида $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{B}^{\mathrm{T}} & \mathbf{D}^{\mathrm{T}} \end{pmatrix}$ и $\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & -\mathbf{D}^{\mathrm{T}} \end{pmatrix}$.

Определение 3. Матрица **T** порядка $n = 4t - 2$, где $n + 1$ — простое число или его степень, — это критская матрица с уровнями блоков 1, $-b$, $d = 0$ (на диагонали), где $b = \frac{\upsilon - 1}{\upsilon + \sqrt{4\upsilon - 3}}$, $\upsilon = n/2 = 2t - 1$ — размер блока.

Определение 4. Матрица **T** порядка $n = 4t - 4$, где $n + 1$ — простое число или его степень, — это критская матрица с уровнями блоков 1, $-b$, $d = \frac{2}{3 + \sqrt{2n + 1}}$ (на диагонали), где $b = 1 - 2d$.

Значения уровней элементов во всех четырех отмеченных определениями случаях следует непосредственно из условия ортогональности и инварианта узора.

Матрицы **H** и **C** порядков $p^m + 1$, где p — простое число, связаны взаимно однозначными преобразованиями с матрицами **O** и **T**. При вычислении их в поле $GF(p^m)$ циклические блоки **A** = **D**, **B** предстают в наиболее экономном виде.

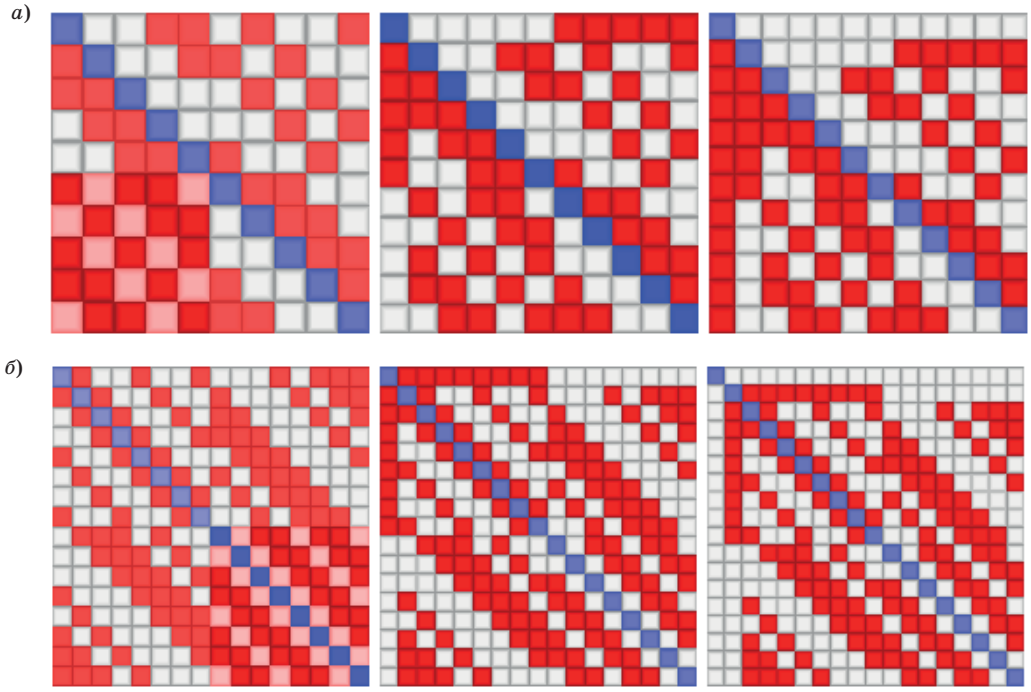
Взаимные переходы дизайнов (узоров) критских матриц показаны для кососимметричного (по знакам) и симметричного случаев на рис. 1, *а* и *б* соответственно. Здесь приведены их портреты — графические представления в цвете.

В первой кососимметричной по **A** версии узора вторая (парная) матрица **B** симметрична, а в симметричной версии — кососимметричной (по знакам) является ее первая строка, в силу четности размера она состоит из инвертированных по знаку половинок. Вторая половинка реверсирована. Именно этот важный инвариант структуры навязывается арифметикой полей Галуа, именно он отвечает за сопровождение порядков матриц простыми числами и их степенями.

Данное обстоятельство вскрыл еще основоположник использования полей Пэли, но в его время (30-е годы XX века) использовалось приведение матиц Адамара к циклическому блоку с одинарной каймой [13] (матриц Белевича тогда еще не было). Эта форма неустойчива к показателю степени простого числа. Для кратных простых чисел основа (core) матриц разваливается на мультиблоки. Бициклический характер основы освобождает нас от погружения в мультиблочные структуры, значительно упрощая подход Пэли.

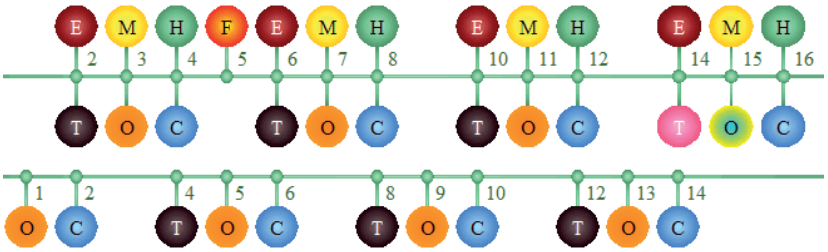
Различие между матрицами Одина и Тени с ранее описанными матрицами Мерсенна и Эйлера [12] фундаментально. Малозначимая на порядках простых чисел замена 0 на 1 на диагонали открывает возможность циклическим смещением блоков **A** и **B** бициклов Эйлера *восстанавливать ортогональность* при отсутствии поля, однозначно связанного с указанными симметриями. Таким образом, помимо матриц, сопровождающих простые числа и их степени, появляются матрицы Эйлера **E** порядков $4t - 2$ и Мерсенна **M** составных порядков $4t - 1$, которые можно найти либо переборами, либо оптимизацией детерминанта [9].

Хорошо известная из литературы невозможность в *рамках комбинаторной теории* [1, 4, 5] доказать наличие матриц Адамара (не встречая принципиальных возражений) не означает, что альтернативная применению техники полей логика поиска экстремумов [9] в чем-либо ущербна. Не менее хорошо известно, что матрицы максимума детерминанта сопровождают все порядки, независимо от их составного характера. На приведенной на рис. 2 диаграмме заметны длинные цепочки матриц **E-M-H-(F)** и короткие вида **T-O-C**.



■ ***Рис. 1.*** Взаимные переходы кососимметричных (*а*) и симметричных (*б*) дизайнов матриц **T**, **O**, **C**

■ ***Fig. 1.*** Mutual transitions of skew-symmetric (*а*) and symmetric (*б*) designs of matrices **T**, **O**, **C**



■ ***Рис. 2.*** Диаграмма цепочек матриц
■ ***Fig. 2.*** Diagram of matrix chains

Матрицы **H** и **C** сосуществуют на порядках $4t$. Некоторые матрицы Адамара сопровождают синхронизированные с последовательностью чисел Ферма критские матрицы **F** порядков $4t + 1$ [9, 12].

Из диаграммы следует, например, что можно находить все четыре разновидности обобщенных критских матриц **E-M** и **T-O** методом сверху (отделением каймы), а не снизу (использованием поля) для составного порядка 15, отвечающего матрицам **H** и **C** порядка 16. В таких случаях критские матрицы будут лишены не существенных для их поиска иным путем инвариантов — у них не будет гарантируемых полем взаимных циклических симметрий их блоков.

Можно показать связь проблемы существования и поиска матриц Адамара с теоремой Гаусса о гарантированном разложении простых и составных чисел на три фигурных числа, отмечаемом еще Ферма. Простое по реализации отделение каймы открывает возможность находить матрицы Одина и Тени (как и матрицы Мерсенна и Эйлера) не снизу, переборами (алгоритмы работы в поле [13, 14] тоже относятся к комбинаторным процедурам) или оптимизацией [8, 9, 15], а сверху, отделением каймы от структур, параметры которых и блочное строение мы здесь описали.

Таким образом, уязвимой является лишь цепочка нижних троиц симметричных матриц **T-O-C** (см. рис. 2), связанных с особенностями разложения чисел, соответствующих порядкам матриц **O**, на суммы двух квадратов. Матрицы могут отсутствовать по двум причинам: либо такой порядок неразложим в указанном смысле, либо он является составным числом, приводящим к мультиблочности. Если бы не последнее обстоятельство, матрицы Белевича порядков 66 и 86 и т. п. были бы давно найдены. Но они до сих пор не известны [1]. Отмеченные матрицы пополняют новыми представителями состав ортогональных матриц семейства Адамара с выделенными чертами симметрии или кососимметрии в рамках исследований, которые ведутся в настоящее время [16–18].

Симметрии матриц Адамара

Матрицы Адамара — это матрицы порядков $n = 4v$, которые традиционно делят на 4, выделяя характерный размер блоков **A**, **B**, **C**, **D** в виде $v = n/4$. Условие ортогональности дает квадратичное уравнение связи $w^2 + x^2 + y^2 + z^2 = n$, регламентирующее число –1 в них: $k_1 = (v - w)/2$, $k_2 = (v - x)/2$, $k_3 = (v - y)/2$, $k_4 = (v - z)/2$.

Для матриц порядков, идущих с шагом 8: $n = 4 + 8t = 4(2t + 1)$, размер блока $v = 2t + 1$ — нечетное число. Поэтому у кососимметричных с точностью до диагонали $\mathbf{A} - \mathbf{I} = (\mathbf{I} - \mathbf{A})^{\mathrm{T}}$ матриц **A** число $k_1 = (v - 1)/2$, т. е. $w = 1$, что сразу же дает уравнение сферы $x^2 + y^2 + z^2 = n - 1$. Для симметричного варианта решения $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$ и $\mathbf{B} = \mathbf{C}$, поменяв местами обозначения w и x, связав свободную переменную x с первой матрицей, имеем $w = y$, что приводит к уравнению сфероида $x^2 + 2y^2 + z^2 = n$.

Разрешимостью в целых числах уравнений сферы и сфероида занимались Гаусс и Лиувилль, сводившие заменой $x^2 = 8T_x + 1$, $y^2 = 8T_y + 1$, $z^2 = 8T_z + 1$ уравнения к линейному виду: $T_x + T_y + T_z = t$ и $T_x + 2T_y + T_z = t$, где t, как и ранее, — натуральное число, задающее номер матрицы в отмеченной числовой последовательности. По теореме Гаусса, любое целое число разрешимо не более чем тремя треугольными числами, т. е. числами, взятыми из последовательности сумм чисел 0, 1, 3, 6, 10 и т. п. (аддитивный факториал). Лиувилль распространил это правило на второе линейное уравнение, близкое к уравнению Гаусса по смыслу.

Таким образом, симметричные и кососимметричные матрицы Адамара сосуществуют на всех порядках, идущих с шагом порядка 8. Разновидность правила Сильвестра $\begin{pmatrix} \mathbf{H} & \mathbf{H} \\ \mathbf{H} & -\mathbf{H} \end{pmatrix}$ и $\begin{pmatrix} \mathbf{H} & \mathbf{H} \\ -\mathbf{H}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} \end{pmatrix}$ позволяет распространить свойство симметрии и кососимметрии на удвоенные порядки. Следовательно, среди матриц Адамара нет такого порядка, на котором нельзя найти две отмеченные симметрии.

Заключение

В настоящей работе свойство симметрии изучается при делении матрицы Адамара на две каймы и соответствующие делениям каймы большие по размеру блоки. При этом оказывается, что симметричные признаки присущи и критским матрицам, которые образуются ортогонализацией основы (core) матриц Адамара при последовательном отделении первой и второй каймы. Поскольку значения элементов матрицы перестают быть целочисленными, понимание симметрии изменяется — мы отслеживаем знаки, а не плавающие при сечении матрицы значения элементов. Разумеется, симметрии матриц, если они есть, сохраняются при выделении основы сверху для любых порядков.

Этим и объясняется наличие матриц Эйлера, не обязательно бициклических, но симметричных или кососимметичных на порядках, на два меньших порядков матриц Адамара. Однако деление может быть тоньше, когда мы обращаем внимание на двоякосимметричные матрицы Одина и Тени, являющиеся основами, в том числе, и конференц-матриц. В этом случае характерные бициклы сопровождают не все порядки, а только равные простым числам и их степеням. Ранее такие критские матрицы не выделялись, и их уровни и симметрии не описывались. Это дает право говорить о новом семействе матриц, позволяющих понять глубже симметрии матриц Адамара и конференц-матриц, основой которых они являются.

Благодарности

Мы выражаем благодарность за многолетнюю помощь и поддержку профессорам Дженнифер Себерри и Драгомиру Джоковичу. За помощь в технической работе с рукописью благодарим Т. В. Балонину.

Финансовая поддержка

Статья подготовлена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, соглашение № FSRF-2020-0004.

Литература

1. **Colbourn C. J., Dinitz J. H.** *Handbook of Combinatorial Designs*. Second Ed. Chapman and Hall/CRC, 2007. 967 p.
2. **Silvester J. J.** Thoughts on inverse orthogonal matrices, simultaneous sign successions, and tessellated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers. *Philosophical Magazine*, 1867, no. 34, pp. 461–475.
3. **Hadamard J.** Résolution d'une question relative aux déterminants. *Bulletin des sciences mathématiques*, 1893, vol. 17, pp. 240–246.
4. **Jennifer S., Yamada M.** *Hadamard Matrices: Constructions using Number Theory and Linear Algebra*. Wiley, 2020. 384 p.
5. **Craigen R.** Hadamard Matrices and Designs. In: *CRC Handbook of Combinatorial Designs*. C. J. Colbourn and J. H. Dinitz eds. CRC Press, 1996. Pp. 229–516.
6. **Kharaghani H., Tayfeh-Rezaie B. A.** Hadamard matrix of order 428. *Journal of Combinatorial Designs*, 2005, vol. 13, pp. 435–440.
7. **Balonin N. A., and Seberry J.** Remarks on extremal and maximum determinant matrices with real entries ≤ 1. *Информационно-управляющие системы*, 2014, № 5, с. 2–4.
8. **Mohan M. T.** *p*-almost Hadamard matrices and λ-planes. *Journal of Algebraic Combinatorics*, 2020, 20 p. ttps://doi.org/10.1007/s10801-020-00991-y
9. **Балонин Н. А., Сергеев М. Б.** Матрицы локального максимума детерминанта. *Информационно-управляющие системы*, 2014, № 1, с. 2–15.
10. **Mohan M. T.** On some *p*-almost Hadamard matrices. *Operators and Matrices*, 2019, vol. 13, no. 1, pp. 253–281. doi:10.7153/oam-2019-13-17
11. Balonin N. A., Đoković D. Ž. Conference matrices from Legendre C-pairs. *Информационно-управляющие системы*, 2020, № 4, с. 2–10. doi:10.31799/1684-8853-2020-4-2-10
12. **Балонин Н. А., Сергеев М. Б.** Как гипотезе Адамара помочь стать теоремой. Ч. 1. *Информационно-управляющие системы*, 2018, № 6, с. 2–13.
13. **Paley R. E. A. C.** On orthogonal matrices. *Journal of Mathematics and Physics*, 1933, vol. 12, pp. 311–320.
14. **Балонин Н. А., Сергеев А. М., Синицына О. А.** Алгоритмы конечных полей и групп поиска ортогональных последовательностей. *Информационно-управляющие системы*, 2021, № 4, с. 2–16.
15. **Wen Z., Yin W.** A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, Ser. A, 2013, vol. 142, pp. 397–434. https://doi.org/10.1007/s10107-012-0584-1
16. **Holzmann W. H., Kharaghani H., Tayfeh-Rezaie B.** Williamson matrices up to order 59. *Designs, Codes and Cryptography*, 2008, vol. 46, iss. 3, pp. 343–352.
17. **Awyzio G., Seberry J.** *On Good Matrices and Skew Hadamard Matrices*. 2015. 15 p. https://www.researchgate.net/publication/285233232_On_Good_Matrices_and_Skew_Hadamard_Matrices (дата обращения: 12.11.2021).
18. **Acevedo S., Dietrich H.** New infinite families of Williamson Hadamard matrices. *Australian Journal of Combinatorics*, 2019, vol. 73, iss. 1, pp. 207–219.

UDC 519.614
doi:10.31799/1684-8853-2022-1-2-7

Odin and Shadow Cretan matrices accompanying primes and their powers

N. A. Balonin[a], Dr. Sc., Tech., Professor, orcid.org/0000-0001-7338-4920, korbendfs@mail.ru
M. B. Sergeev[a], Dr. Sc., Tech., Professor, orcid.org/0000-0002-3845-9277
[a]Saint-Petersburg State University of Aerospace Instrumentation, 67, B. Morskaia St., 190000, Saint-Petersburg, Russian Federation

Introduction: Cretan matrices — orthogonal matrices, consisting of the elements 1 and $-b$ (real number), are an ideal object for the visual application of finite-dimensional mathematics. These matrices include, in particular, the Hadamard matrices and, with the expansion of the number of elements, the conference matrices. The most convenient research apparatus is to use field theory and multiplicative Galois groups, which is especially important for new types of Cretan matrices. **Purpose:** To study the symmetries of the Cretan matrices and to investigate two new types of matrices of odd and even orders, distinguished by symmetries, respectively, which differ significantly from the previously known Mersenne, Euler and Fermat matrices. **Results:** Formulas for levels are given and symmetries of new Cretan matrices: Odin bicycles (with a border) of orders $4t - 1$ and $4t - 3$ and shadow matrices of orders $4t - 2$ and $4t - 4$ are described. For odd character sizes equal to prime numbers and powers of primes, the existence of matrix symmetries of special types, doubly symmetric, consisting of skew-symmetric (with respect to the signs of elements) and symmetric cyclic blocks, is proved. It is shown that the previously distinguished Cretan matrices are their special case: Mersenne matrices of orders $4t - 1$ and Euler matrices of orders $4t - 2$ existing in the absence of symmetry for all selected orders without exception. **Practical relevance:** Orthogonal sequences and methods of their effective finding by the theory of finite fields and groups are of direct practical importance for the problems of noise-immune coding, compression and masking of video information.

Keywords — Hadamard matrices, Belevich matrices, Cretan matrices, finite fields, matrix symmetries.

For citation: Balonin N. A., Sergeev M. B. Odin and Shadow Cretan matrices accompanying primes and their powers. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 2–7 (In Russian). doi:10.31799/1684-8853-2022-1-2-7

Financial support

The article was prepared with the financial support of the Ministry of Science and Higher Education of the Russian Federation, agreement No. FSRF-2020-0004.

References

1. Colbourn C. J., Dinitz J. H. *Handbook of Combinatorial Designs*. Second Ed. Chapman and Hall/CRC, 2007. 967 p.
2. Silvester J. J. Thoughts on inverse orthogonal matrices, simultaneous sign successions, and tessellated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers. *Philosophical Magazine*, 1867, no. 34, pp. 461–475.
3. Hadamard J. Résolution d'une question relative aux déterminants. *Bulletin des sciences mathématiques*, 1893, vol. 17, pp. 240–246 (In French).
4. Jennifer S., Yamada M. *Hadamard Matrices: Constructions using number theory and linear algebra*. Wiley, 2020. 384 p.
5. Craigen R. *Hadamard matrices and designs*. In: *CRC Handbook of Combinatorial Designs*. C. J. Colbourn and J. H. Dinitz eds. CRC Press, 1996. Pp. 229–516.
6. Kharaghani H., Tayfeh-Rezaie B. A. Hadamard matrix of order 428. *Journal of Combinatorial Designs*, 2005, vol. 13, pp. 435–440.
7. Balonin N. A., and Seberry J. Remarks on extremal and maximum determinant matrices with real entries ≤ 1. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2014, no. 5, pp. 2–4.
8. Mohan M. T. *p*-almost Hadamard matrices and λ-planes. *Journal of Algebraic Combinatorics*, 2020. 20 p. ttps://doi.org/10.1007/s10801-020-00991-y
9. Balonin N. A., Sergeev M. B. Local maximum determinant matrices. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2014, no. 1, pp. 2–15 (In Russian).
10. Mohan M. T. On some *p*-almost Hadamard matrices. *Operators and Matrices*, 2019, vol. 13, no. 1, pp. 253–281. doi:10.7153/oam-2019-13-17
11. Balonin N. A., Đoković D. Ž. Conference matrices from Legendre C-pairs. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 4, pp. 2–10. doi:10.31799/1684-8853-2020-4-2-10
12. Balonin N. A., Sergeev M. B. Helping Hadamard conjecture to become a theorem. Part 1. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2018, no. 6, pp. 2–13 (In Russian). doi:10.31799/1684-8853-2018-6-2-13
13. Paley R. E. A. C. On orthogonal matrices. *Journal of Mathematics and Physics*, 1933, no. 12, pp. 311–320.
14. Balonin N. A., Sergeev A. M., Sinitsyna O. I. Finite field and group algorithms for orthogonal sequence search. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2021, no. 4, pp. 2–16 (In Russian). doi:10.31799/1684-8853-2021-4-2-17
15. Wen Z., Yin W. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, Ser. A, 2013, vol. 142, pp. 397–434. https://doi.org/10.1007/s10107-012-0584-1
16. Holzmann W. H., Kharaghani H., Tayfeh-Rezaie B. Williamson matrices up to order 59. *Designs, Codes and Cryptography*, 2008, vol. 46, iss. 3, pp. 343–352.
17. Awyzio G., Seberry J. *On Good Matrices and Skew Hadamard Matrices*. 2015. 15 p. Available at: https://www.researchgate.net/publication/285233232_On_Good_Matrices_and_Skew_Hadamard_Matrices (accessed 12 November 2021).
18. Acevedo S., Dietrich H. New infinite families of Williamson Hadamard matrices. *Australian Journal of Combinatorics*, 2019, vol. 73, iss. 1, pp. 207–219.

УДК 004.056
doi:10.31799/1684-8853-2022-1-8-18

Научные статьи
Articles

Определение психического состояния пользователей социальной сети Reddit на основе методов машинного обучения

А. А. Браницкий[а,б], *канд. техн. наук, старший научный сотрудник, orcid.org/0000-0003-3104-0622, branitskiy@comsec.spb.ru*
Я. Д. Шарма[в], *студент, orcid.org/0000-0003-2491-0167*
И. В. Котенко[а,б], *доктор техн. наук, профессор, orcid.org/0000-0001-6859-7120*
Е. В. Федорченко[а,б], *канд. техн. наук, старший научный сотрудник, orcid.org/0000-0001-6707-9153*
А. В. Красов[б], *канд. техн. наук, доцент, orcid.org/0000-0002-9076-6055*
И. А. Ушаков[б], *канд. техн. наук, доцент, orcid.org/0000-0002-6988-9261*
[а]*Санкт-Петербургский Федеральный исследовательский центр РАН, 14-я линия В. О., 39, Санкт-Петербург, 199178, РФ*
[б]*Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, Большевиков пр., 22-1, Санкт-Петербург, 193232, РФ*
[в]*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», Профессора Попова ул., 5, Санкт-Петербург, 197376, РФ*

Введение: *диагностирование психических заболеваний представляет собой сложный процесс, который включает проведение диалоговых бесед, анализ поведения обследуемого и прохождение им специализированных тестов. На успешное решение данной задачи может влиять как отсутствие знаний и опыта психолога, так и наличие противоречивых или неполных исходных данных со стороны пациента. Для устранения последнего недостатка разрабатываются экспертные или интеллектуальные системы.* ***Цель:*** *разработать методику определения психического состояния пользователей социальной сети.* ***Результаты:*** *с помощью методов машинного обучения разработана методика, предназначенная для определения типа психического состояния пользователей социальной сети. Новизна предлагаемой методики заключается в наличии двухшаговой процедуры предварительной обработки текста и построении нескольких наборов признаков, описывающих эмоциональное настроение пользователей социальной сети на уровне публикуемых ими сообщений. В качестве исходных данных привлекались текстовые сообщения пользователей социальной сети Reddit. В методике выделяются три этапа: 1) сбор данных, 2) предварительная обработка данных, 3) разметка постов и построение признаков. Оценка функционирования программного средства, построенного на основе данной методики, проводилась по четырем показателям: достоверность, точность, полнота и F-мера. Наилучшие результаты демонстрирует ансамбль, построенный на основе подхода One-vs-Rest, где в качестве базовых решателей выступают линейные машины опорных векторов.* ***Практическая значимость:*** *результаты исследования могут применяться при построении вспомогательных систем, которые направлены на поддержку принятия решений специалистами-психологами при определении психических нарушений.*

Ключевые слова *— машинное обучение, социальная сеть, психические нарушения, эмоциональное настроение, машина опорных векторов, сверточная нейронная сеть.*

Для цитирования: Браницкий А. А., Шарма Я. Д., Котенко И. В., Федорченко Е. В., Красов А. В., Ушаков И. А. Определение психического состояния пользователей социальной сети Reddit на основе методов машинного обучения. *Информационно-управляющие системы*, 2022, № 1, с. 8–18. doi:10.31799/1684-8853-2022-1-8-18
For citation: Branitskiy A. A., Sharma Y. D., Kotenko I. V., Fedorchenko E. V., Krasov A. V., Ushakov I. A. Determination of the mental state of users of the social network Reddit based on machine learning methods. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 8–18 (In Russian). doi:10.31799/1684-8853-2022-1-8-18

Введение

Современные интернет-платформы, такие как социальные сети, предоставляют своим пользователям множество функций, которые позволяют им обсуждать интересующие их темы, делиться друг с другом графической и текстовой информацией, выражать собственные мнения и эмоции. Благодаря наличию таких открытых социальных сетей, как Facebook, Twitter, Instagram, Snapchat, Reddit, люди стали часто использовать их для создания сообществ и обсуждения общих вопросов. Рост популярности таких сетевых сервисов приводит к генерации огромного объема данных, которые включают анкетные данные пользователей, содержимое постов и сообщений, комментарии, отметки о количестве просмотров, загруженные аудио- и графические файлы. С другой стороны, в социальных сетях могут распространяться ложные новости, присутствовать пропаганда нездорового образа жизни и вредных привычек, содержаться призывы к выполнению

противозаконных действий. Наличие такой информации негативно влияет на поведение пользователей и приводит к нарушениям в их психическом здоровье.

Для оценки психического состояния исследуемого человека специалисты используют текстовую информацию, полученную в результате диалога с ним. Результаты обследования являются конфиденциальными, в связи с чем ими нельзя воспользоваться при проведении открытых исследований. В то же время такая социальная сеть, как Reddit, предоставляет возможность построения и организации сообществ (subreddits), внутри которых ведется открытое обсуждение вопросов, к примеру, связанных с настроением или психическим здоровьем участников определенной группы. Кроме того, извлекаемая из постов социальной сети информация может отражать эмоциональное состояние автора (на основе наличия определенных ключевых слов и последовательностей знаков препинания). Поэтому социальную сеть можно рассматривать в качестве платформы, подходящей для сбора и анализа информации о психическом состоянии ее пользователей.

Правильность и своевременность определения типа и степени психического расстройства позволяет сформировать корректный план лечения для пациента и вовремя предотвратить у него развитие возможных болезней. Успех решения этой задачи во многом зависит как от степени квалификации психолога, так и от уровня откровенности обследуемого пациента. В то же время в связи с многообразием и популярностью интернет-ресурсов (форумов, каналов в мессенджерах, сообществ в социальных сетях) пользователям становится проще излагать свои мысли, выражать эмоции и консультироваться со специалистами в анонимном формате, не прибегая к реальному диалогу. В связи с этим актуальной является задача, заключающаяся в разработке методик и информационных систем, предназначенных для анализа открыто публикуемых в социальных сетях постов и направленных на обнаружение отклонений в психическом состоянии их авторов.

Анализ релевантных работ

Выявление нарушений (депрессии, тревоги, стресса) в психическом состоянии человека является важной задачей, решению которой посвящено множество работ. С учетом массовой доступности Интернета социальные сети могут рассматриваться как платформа для сбора информации об определении психического состояния их пользователей. К примеру, в работе [1] выполнялось обнаружение стрессового состояния у пользователей социальной сети Twitter. Предложенный подход был направлен на определение текущего статуса и возможных изменений в поведении подростков, находящихся под психологическим давлением. Аналогичные задача и социальная сеть были рассмотрены в [2], где авторы добавили статистические атрибуты, которые были получены в результате обработки сообщений, опубликованных в течение недели. Отмечается, что использование расширенного таким образом набора признаков позволяет анализировать совокупность событий, возможно, послуживших причиной нарушения психического состояния у адресанта сообщения.

Состояние депрессии является распространенной формой нарушения психического здоровья, обнаружению которого посвящено исследование [3]. Авторы этой статьи отмечают возможную корреляцию между депрессивным состоянием пользователя и наличием эмоциональных слов в его сообщениях. В [4] обнаружение депрессивного состояния пользователей Twitter выполнялось на основе анализа временных рядов, представляющих собой последовательность параметров активности каждого пользователя в течение года. Отмечается, что у людей, страдающих депрессивными расстройствами, наблюдается низкая социальная активность, преобладание негативных эмоций и завышенное внимание к своей личности. В [5] при решении аналогичной задачи был определен минимальный период (длительностью два месяца), в течение которого построенный набор статистических признаков, описывающих активность пользователей Twitter, доставляет наибольшую точность (69 %) достоверного определения депрессивного состояния.

В отличие от Twitter, в социальной сети Facebook допустимый размер публикуемых постов практически не ограничен, что позволяет извлечь больший объем анализируемой информации из каждого поста. В [6] данная социальная сеть послужила источником для проверки гипотезы о сезонности проявления депрессии. Степень тяжести этого состояния может колебаться в зависимости как от перемен в личной жизни, так и от факторов окружающей среды. Для количественного прогнозирования этой величины авторы разработали регрессионную модель, при обучении которой использовались результаты опроса пользователей и данные, касающиеся обновления статуса в их профилях. Согласно построенной модели уровень прогнозируемого депрессивного состояния у пользователей зимой оказывался выше, чем летом.

Кроме текстовой информации, фотографии, размещенные в профиле социальной сети Facebook, также могут быть информативными при определении особенностей характера человека [7]. Было отмечено, что существует прямая зависимость между

наличием депрессии в поведении человека и преобладанием темных тонов и замкнутых пространств на его фотографии. Похожая задача, связанная с выявлением психических заболеваний на основе анализа изображений, решается в [8], однако в ней рассматривается другая социальная сеть, а именно Instagram. В результате проведения исследования было выявлено наличие корреляции между цветовыми схемами в изображении и депрессивным состоянием ее автора. В данном исследовании извлекаемые признаки были разбиты по двум типам. Для построения признаков первого типа использовались показатели, связанные с активностью пользователя и его подписчиков (количество комментариев и отметок «нравится» для каждого поста). Признаки второго типа подразумевали детальный анализ изображений (количество человеческих лиц на изображении; средние значения показателей пикселей: оттенка, насыщенности и яркости; наличие Instagram-фильтра на изображении). Явное задание набора вычисляемых признаков, как это показано в [8], подразумевает извлечение из изображений таких индикаторов, которые кажутся с экспертной точки зрения репрезентативными для выявления депрессии. С другой стороны, существуют исследования, в которых процесс формирования этих признаков перекладывается непосредственно на модель машинного обучения. Так, в [9] исследовалась применимость двух типов глубоких нейронных сетей для анализа сообщений из социальной сети Reddit и их классификации в соответствии с 11 типами психических заболеваний. Для увеличения точности прогнозирования таких заболеваний в [10] предлагается использовать несколько бинарных классификаторов, каждый из которых предназначен для выявления наличия только одного психического расстройства и построен на основе XGBoost или сверточной нейронной сети. Другой вариант повышения качества анализа постов в социальных сетях может включать использование современных моделей обработки естественного языка, например BERT [11]. Применение этой модели и ее модификации в виде RoBERTa [12] рассматривается в [13], где отмечается их превосходство над сетью с долгой краткосрочной памятью. При этом проведенные для социальной сети Reddit эксперименты выполнялись таким образом, что в качестве входных данных анализировались как посты и заголовки по отдельности, так и их объединение.

Дополнительный прирост производительности классификаторов возможен за счет построения разнородного списка признаков, покрывающих анализ комментариев, изображений и профиля на странице пользователя социальной сети. В [14] с использованием такого списка признаков исследовалась применимость нескольких типов нейронных сетей. В рамках задачи прогнозирования подверженности пользователей социальных сетей деструктивным воздействиям наилучшие результаты были достигнуты при помощи нейронной сети с тремя скрытыми слоями.

В отличие от представленных работ, разработанная методика отличается наличием двухшаговой процедуры предварительной обработки текста, а также возможностью построения нескольких наборов признаков, описывающих эмоциональное настроение пользователей социальной сети на уровне публикуемых ими сообщений.

Методика определения психического состояния пользователей социальной сети Reddit

Разработанная методика определения психического состояния пользователей социальной сети Reddit включает три этапа. Первый этап — сбор данных. С этой целью использовался общедоступный интерфейс прикладного программирования API Pushshift Reddit Dataset [15]. В Reddit пользователи объединяются в сообщества со схожими интересами и убеждениями. В рамках одного сообщества его участники могут отправлять друг другу сообщения, комментировать сообщения других людей и голосовать за или против. После сбора данных из социальной сети необходимо провести их предварительную обработку, выполняющуюся на втором этапе методики. Для этого используется двухшаговая процедура, включающая фильтрацию сообщений и обработку их содержимого. Наконец, на третьем этапе выполняется разметка постов и извлечение признаков. При присвоении постам меток психических расстройств их авторов использовались следующие шесть классов: депрессия, тревога, членовредительство, стресс, гнев и норма. В качестве классификаторов использовались следующие два типа моделей машинного обучения:

1) линейные классификаторы: линейная машина опорных векторов (МОВ) и ансамбли, использующие в качестве базовых классификаторов МОВ и построенные на основе одной из двух стратегий комбинирования One-vs-One (OvO) или One-vs-Rest (OvR);

2) текстовые классификаторы: fastText и сверточная нейронная сеть (СНС).

Сбор данных

Каждая запись в загруженном наборе данных содержит семь полей, включая автора и заголовок сообщения, а также время его создания (табл. 1). Наиболее информативным среди входных данных является поле, обозначенное Selftext.

■ ***Таблица 1***. Список полей с их описанием и типом данных в наборе данных

■ ***Table 1***. List of fields with their description and data type within the dataset

Поле	Описание	Тип данных
Author	Имя пользователя, который опубликовал запись	Строка
Created_utc	Метка времени публикации записи (в формате POSIX-времени)	Целое число
Title	Заголовок публикации	Строка
Selftext	Содержимое публикации	Строка
Score	Рейтинг публикации, вычисляемый как разность количества голосов «за» и количества голосов «против»	Целое число
URL	URL-адрес публикации	Строка
Subreddit	Название сообщества, в котором размещена публикация	Строка

В Reddit нет ограничений на размер публикуемых сообщений, поэтому пользователи могут свободно выражать свои мысли и идеи, используя неограниченное количество слов. В зависимости от состояния психического здоровья участников можно выделить несколько сообществ. В настоящем исследовании рассмотрены такие классы психических нарушений, как депрессия, тревога, членовредительство, гнев и стресс. В табл. 2 для каждого из этих классов и класса «норма» приводится краткое описание соответствующих им сообществ, а также перечисляются сведения о количестве участников и сообщений. В результате анализа собранных данных было выявлено, что наибольшая доля постов относится к классу «депрессия». Также для этого класса характерно наибольшее количество участников. Рассматриваемые данные были собраны в период с января 2018 года по апрель 2021 года.

Предварительная обработка данных

Этап предобработки данных заключается в приведении исходных данных к определенному формату и удалении избыточных данных. Данный этап выполняется в два шага.

■ ***Таблица 2***. Reddit-сообщества, связанные с вопросами психического здоровья

■ ***Table 2***. Reddit-communities related with questions of mental health

Сообщество и класс	Описание сообщества	Количество участников	Количество постов
r/depression, депрессия	Поддерживающее сообщество, которое помогает любому, кто борется с депрессией, и предоставляет открытое пространство для разговоров и обсуждений	757 тыс.	1,2 млн
r/depressed, депрессия	Сообщество людей, страдающих депрессией или находящихся в депрессивном состоянии	74 тыс.	30 тыс.
r/Anxiety, тревога	Сообщество, в котором проводятся обсуждения, связанные с тревожными расстройствами	455 тыс.	410 тыс.
r/Anxietyhelp, тревога	Сообщество, в котором участники делятся статьями, видео- и текстовыми сообщениями в блогах, чтобы справляться с тревогой	91 тыс.	24 тыс.
r/selfharm, членовредительство	Сообщество, в котором участники обсуждают членовредительство и его аспекты	71 тыс.	141 тыс.
r/SuicideWatch, членовредительство	Сообщество, которое поддерживает людей, размышляющих о самоубийстве, или лиц, подверженных риску самоубийства	274 тыс.	580 тыс.
r/Anger, гнев	Сообщество, в котором участники обсуждают проблемы гнева и способы борьбы с ним	27 тыс.	14 тыс.
r/Stress, стресс	Сообщество, обсуждающее причины стресса и методы управления стрессом	11 тыс.	7.5 тыс.
r/philosophy, норма	Сообщество, в котором участники обсуждают философские вопросы	15,7 млн	174 тыс.
r/AskReddit, норма	Сообщество, в котором участники задают наводящие на размышления вопросы с целью получить на них ответы	32,3 млн	27 млн

Шаг 1.

1. Удаляются сообщения, авторы которых исключились из социальной сети.

2. Удаляются сообщения, у которых часть содержимого (Selftext) недоступна.

3. Удаляются сообщения, имеющие схожее содержимое, но опубликованные в разных сообществах.

4. Удаляются сообщения наименее активных пользователей, имеющих менее 50 опубликованных постов.

5. Удаляются сообщения пользователей, являющихся возможными ботами и имеющих более 5000 постов.

Шаг 2.

1. Заголовок и содержимое поста объединяются в единое сообщение.

2. Удаляются все ссылки и URL-адреса, символы новой строки и табуляции заменяются символом пробела.

3. Сленговые выражения заменяются их полными эквивалентными формами, например, idk — I do not know, ur — your, fam — family.

4. Сокращения заменяются их полными аналогами, например, I'll — I will, let's — let us, couldn't've — could not have.

5. Удаляются все специальные символы из текста.

6. Удаляется запятая из чисел.

7. Все числа в тексте нормализуются путем их замены на текстовую форму этого числа.

8. Эмодзи-символы заменяются на их текстовые обозначения.

9. Исправляются опечатки при помощи python-пакета wordninja.

Статистические сведения об экспериментальном наборе данных после этапа предобработки приведены в табл. 3.

■ ***Таблица 3***. Статистические сведения об экспериментальном наборе данных

■ ***Table 3***. Statistical data about experimental dataset

Класс	Количество постов	Среднее число слов в посте	Среднее число символов в посте
Депрессия	40 003	127.075	618.99
Тревога	10 164	138.257	687.271
Членовредительство	25 734	114.211	550.039
Гнев	9260	210.887	1040.01
Стресс	3954	171.799	867.976
Норма	40 633	47.9625	254.905

Разметка постов и построение признаков

Разметка данных по принадлежности к классам депрессии и тревоги выполнялась при помощи поиска самовыражений [16], указывающих на соответствующее психическое нарушение. Например, выражение «I (was|am) diagnosed with (depression|anxiety)», в котором присутствует местоимение первого лица, может свидетельствовать о наличии такого нарушения. Для классов «членовредительство», «стресс» и «гнев» разметка постов выполнялась аналогичным образом, но поиск осуществлялся только с использованием ключевых слов. Записям, не связанным с психическими нарушениями, была присвоена метка класса «норма».

Процесс построения вектора признаков подразумевает вычисление таких показателей, которые по отдельности или в совокупности будут находиться в корреляционной зависимости от прогнозируемой метки класса. Поэтому нахождение таких показателей требует от исследователя определенных знаний предметной области. В данном исследовании были выделены представленные ниже признаки.

1. Количество эмодзи-символов в публикации (emojiCount). Вычисление этого признака выполнялось перед предварительной обработкой текста. На рис. 1 показано распределение величины emojiCount в зависимости от метки класса.

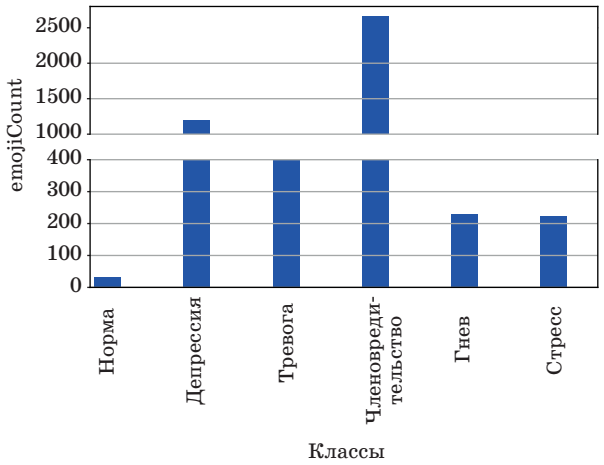
2. Количество местоимений первого и второго лица (firstPropnCount, secondPropnCount). Согласно [17] посты пользователей с психическими нарушениями характеризуются большим количеством личных местоимений. Для подтверждения этой гипотезы была выполнена оценка среднего количества местоимений в сообщении с разбивкой по классам (рис. 2).

3. Количество специальных наречий частотности, таких как абсолютно, постоянно, всегда, никогда, целиком, полностью (absWordCount). В соответствии с [18] пользователи с психическими нарушениями при выражении своих мыслей используют больше абсолютных слов по сравнению со здоровыми людьми. На рис. 3 показана зависимость среднего количества абсолютных слов в сообщении от метки класса.

4. Количество положительных (www.enchantedlearning.com/wordlist/positivewords.shtml) и отрицательных (www.enchantedlearning.com/wordlist/negativewords.shtml) слов (posWordCount, negWordCount). На рис. 4 показана зависимость среднего количества таких слов в сообщении от метки класса.

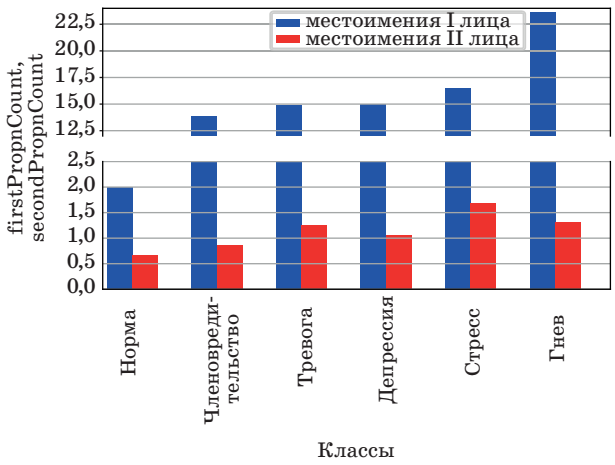
5. Общее количество слов в сообщении (wordCount).

6. Сентимент-оценка сообщения (afinnScore) [19].



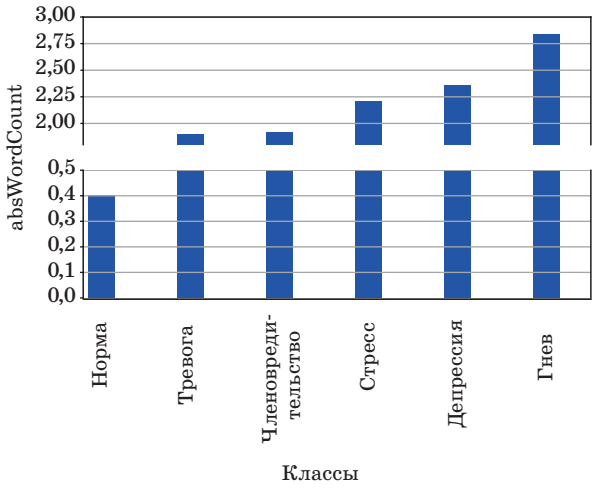
■ ***Рис. 1.*** Распределение величины emojiCount в зависимости от метки класса

■ ***Fig. 1.*** Distribution of emojiCount value in dependence of class label



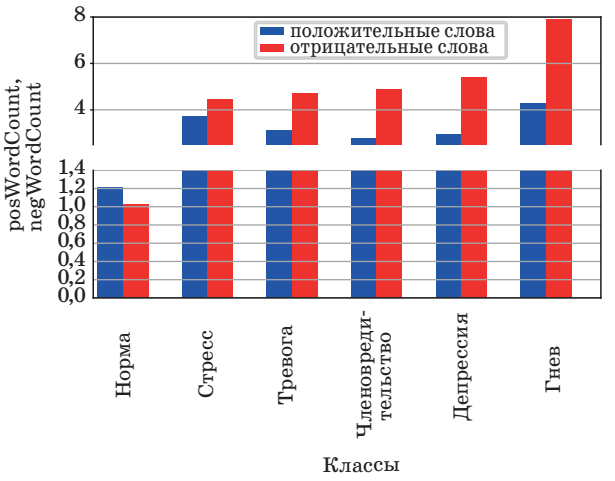
■ ***Рис. 2.*** Зависимость среднего количества местоимений в сообщении от метки класса

■ ***Fig. 2.*** Dependence of average quantity of pronouns within message from class label



■ ***Рис. 3.*** Зависимость среднего количества абсолютных слов в сообщении от метки класса

■ ***Fig. 3.*** Dependence of average quantity of absolute words within message from class label



■ ***Рис. 4.*** Зависимость среднего количества положительных и отрицательных слов в сообщении от метки класса

■ ***Fig. 4.*** Dependence of average quantity of positive and negative words within message from class label

7. Признаки, построенные на основе мешка слов (bowVec).

8. Признаки, построенные на основе TF-IDF (tfidfVec).

Эксперименты

При проведении экспериментов использовались несколько наборов признаков (НП):

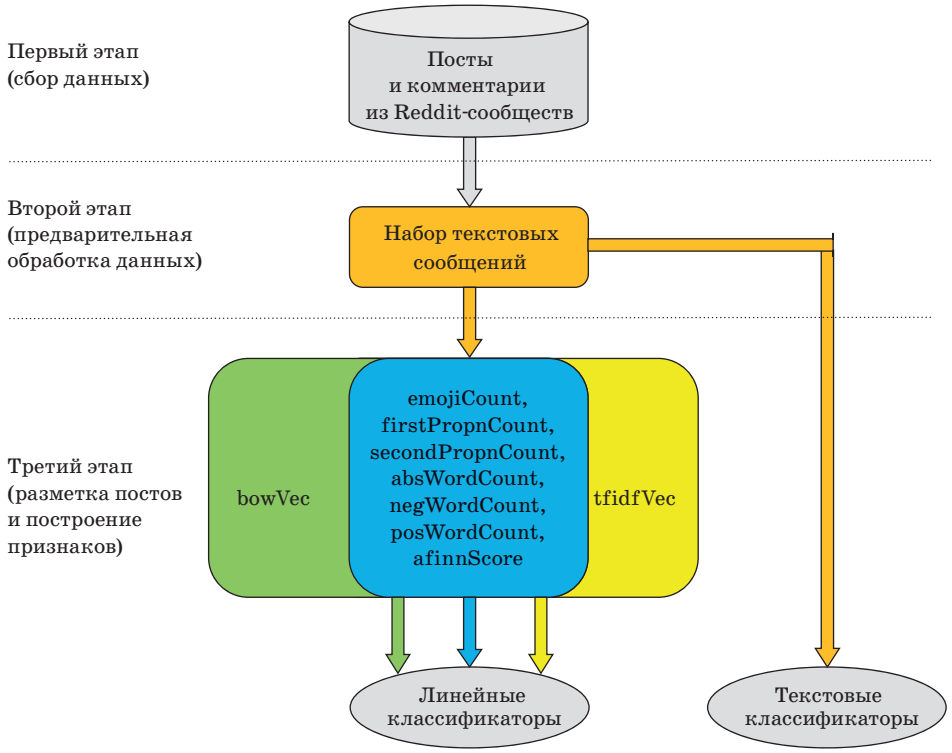
1) I НП — семь признаков (emojiCount, firstPropnCount, secondPropnCount, absWordCount, negWordCount, posWordCount, afinnScore);

2) II НП — восемь признаков (emojiCount, firstPropnCount, secondPropnCount, absWordCount, negWordCount, posWordCount, afinnScore, bowVec);

3) III НП — восемь признаков (emojiCount, firstPropnCount, secondPropnCount, absWordCount, negWordCount, posWordCount, afinnScore, tfidfVec);

4) IV НП — текстовые сообщения.

Схема проведения экспериментов согласно разработанной методике представлена на рис. 5. Обучение линейных классификаторов выполнялось с использованием I, II и III НП, обучение текстовых классификаторов — с использованием только IV НП. Для нахождения оптимального набора гиперпараметров у указанных классификаторов применялся python-модуль GridSearchCV.

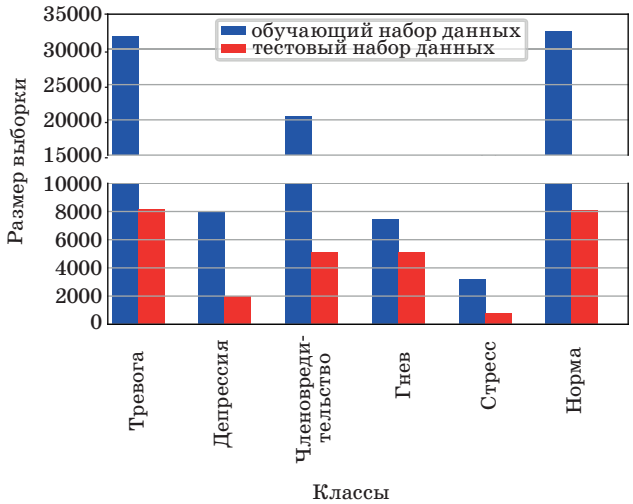


■ ***Рис. 5.*** Схема проведения эксперимента
■ ***Fig. 5.*** Experiment scheme

Для линейных классификаторов наилучшие результаты были достигнуты при использовании III НП.

Разбиение набора данных на обучающую и тестовую выборки выполнялось в отношении 4:1 (рис. 6). При проведении экспериментов использовалась пятиблочная перекрестная проверка [20], что позволило оценить показатели эффективности классификаторов на нескольких дизъюнктных выборках из исходного набора данных. Полученные для каждого классификатора результаты в виде матриц неточностей представлены на рис. 7, *а–д*.

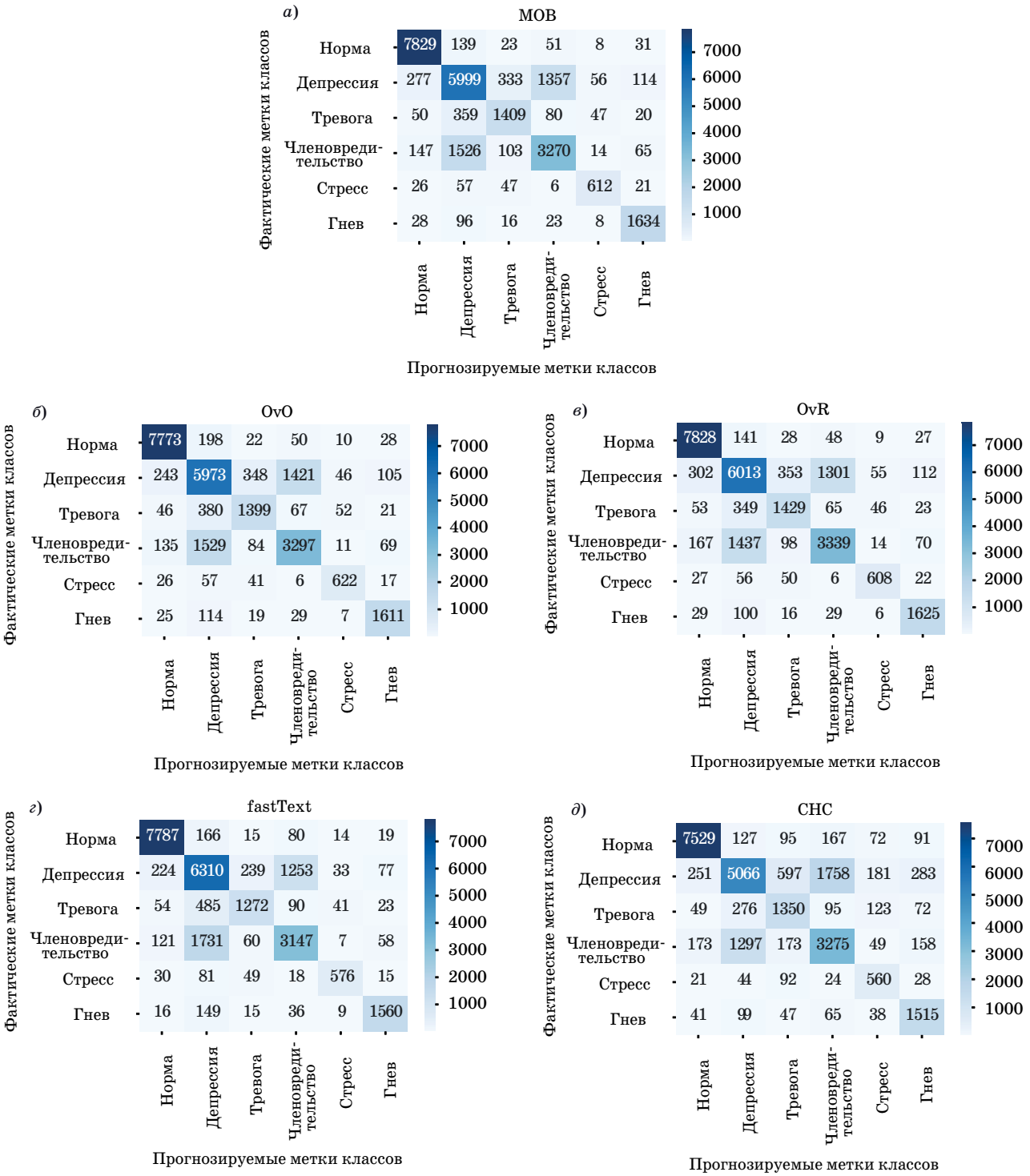
Обобщение полученных результатов представлено в виде табл. 4, которая содержит значения



■ ***Рис. 6.*** Обучающий и тестовый наборы данных
■ ***Fig. 6.*** Training and testing datasets

■ ***Таблица 4.*** Показатели эффективности пяти классификаторов
■ ***Table 4.*** Effectiveness indicators of five classifiers

Классификатор	Достоверность, %	Точность, %	Полнота, %	F-мера, %
MOB	80,19	79,53	79,37	79,42
OvO	79,88	79,61	79,21	79,39
OvR	**80,53**	79,73	**79,62**	**79,65**
fastText	79,86	**80,7**	77,06	78,68
CHC	74,55	68,36	74,13	70,77



■ ***Рис. 7.*** Матрица неточностей для MOB (*а*); OvO (*б*); OvR (*в*); fastText (*г*); CHC (*д*)

■ ***Fig. 7.*** Confusion matrix for SVM (*а*); OvO (*б*); OvR (*в*); fastText (*г*); convolutional neural network (*д*)

достоверности (accuracy) и усредненные по шести классам значения точности (precision), полноты (recall) и F-меры (F-measure), вычисленные для выбранных пяти классификаторов.

Наилучшие результаты в терминах достоверности демонстрирует классификатор OvR. Наибольшее значение показателя точности принадлежит классификатору fastText. В то же время максимальное значение показателя F-меры, являющегося интегральной характеристикой точности и полноты, достигается при помощи классификатора OvR. Полученные результаты

подтверждают достижение поставленной цели и доказывают, что разработанный в данном исследовании список признаков позволяет повысить показатели эффективности классификаторов, обученных для определения психических нарушений на уровне текстовых сообщений, по сравнению с признаками, автоматически формируемыми в случае классификаторов fastText и CHC.

Заключение

В статье описана разработанная методика для определения психического состояния пользователей социальной сети Reddit. В качестве исходных данных рассмотрены текстовые сообщения, публикуемые в качестве постов и комментариев в рамках сообществ данной социальной сети. Для повышения показателей эффективности исследуемых классификаторов машинного обучения предложена двухшаговая процедура предварительной обработки текста и построено несколько наборов признаков, учитывающих эмоциональное настроение пользователей социальной сети на уровне публикуемых ими сообщений. Экспериментальная проверка методики осуществлялась через пятиблочную перекрестную проверку, в результате применения которой наилучшие результаты при определении психических нарушений достигнуты при помощи комбинированного классификатора, построенного на основе подхода One-vs-Rest, где в качестве базовых решателей выступают линейные машины опорных векторов. Как направление дальнейших исследований стоит отметить расширение набора данных за счет сбора данных из других социальных сетей и анализа изображений, а также построение ансамблей, сочетающих разнородные классификаторы.

Финансовая поддержка

Работа выполнена при частичной финансовой поддержке проекта РФФИ 18-29-22034 мк и бюджетной темы 0073-2019-0002.

Литература

1. **Xue Y., Li Q., Jin L., Feng L., Clifton D. A., Clifford G. D.** *Detecting Adolescent Psychological Pressures from Micro-Blog*. Health Information Science. Y. Zhang, G. Yao, J. He, L. Wang, N. R. Smalheiser, X. Yin Eds. Lecture Notes in Computer Science, 2014. Vol. 8423. Pp. 83–94.
2. **Lin H., Jia J., Guo Q., Xue Y., Li Q., Huang J., Cai L., Feng L.** User-level psychological stress detection from social media using deep neural network. *Proc. 22nd ACM International Conference on Multimedia*, 2014, pp. 507–516.
3. **Park M., Chiyoung C., Meeyoung C.** Depressive moods of users portrayed in Twitter. *Proc. HI-KDD*, Beijing, China, ACM, 2012. https://nyuscholars.nyu.edu/ws/files/134720119/depressive_moods_kdd.pdf (дата обращения: 05.08.2021).
4. **De Choudhury M., Gamon M., Counts S., Horvitz E.** Predicting depression via social media. *Proc. of the International AAAI Conference on Web and Social Media*, 2013, AAAI, vol. 7, no. 1, pp. 128–137.
5. **Tsugawa Sh., Kikuchi Y., Kishino F., Nakajima K., Itoh Y., Ohsaki H.** Recognizing depression from twitter activity. *Proc. 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 3187–3196. doi:10.1145/2702123.2702280
6. **Schwartz H., Eichstaedt J., Kern M., Park G., Sap M., Stillwell D., Kosinski M., Ungar L.** Towards assessing changes in degree of depression through Facebook. *Proc. the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, 2014, pp. 118–125. doi:10.3115/v1/W14-3214
7. **Segalin C., Celli F., Polonio L., Kosinski M., Stillwell D., Sebe N., Cristani M., Lepri B.** What your Facebook profile picture reveals about your personality? *Proc. the 25th ACM International Conference on Multimedia*, 2017, pp. 460–468. doi: 10.1145/3123266.3123331
8. **Reece A. G., Christopher M. D.** Instagram photos reveal predictive markers of depression. *EPJ Data Science*, 2017, no. 6, pp. 1–12.
9. **Gkotsis G., Oellrich A., Velupillai S., Liakata M., Hubbard T. J. P., Dobson R. J. B., Dutta R.** Characterisation of mental health conditions in social media using Informed Deep Learning. *Scientific Reports*, 2017, vol. 7, pp. 1–11.
10. **Kim J., Lee J., Park E., Han J.** A deep learning model for detecting mental illness from user content on social media. *Scientific Reports*, 2020, vol. 10, pp. 1–6. doi:10.1038/s41598-020-68764-y
11. **Devlin J., Chang M.-W., Lee K., Toutanova K.** Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. NAACL-HLT*, Minneapolis, Minnesota, 2019, pp. 4171–4186.
12. **Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V.** *Roberta: A robustly optimized BERT pretraining approach*. Preprint, 2019. https://arxiv.org/pdf/1907.11692.pdf (дата обращения: 05.08.2021).
13. **Murarka A., Balaji R., Sushma R.** *Detection and classification of mental illnesses on social media using RoBERTa*. Preprint, 2020. https://arxiv.org/pdf/2011.11226.pdf (дата обращения: 05.08.2021).
14. **Браницкий А. А., Дойникова Е. В., Котенко И. В.** Использование нейросетей для прогнозирования подверженности пользователей социальных сетей

деструктивным воздействиям. *Информационно-управляющие системы*, 2020, № 1, с. 24–33. doi:10.31799/1684-8853-2020-1-24-33
15. **Baumgartner J., Zannettou S., Keegan B., Squire M., Blackburn J.** The Pushshift Reddit dataset. *Proc. the International AAAI Conference on Web and Social Media*, 2020, vol. 14, pp. 830–839.
16. **Losada D. E., Crestani F.** A test collection for research on depression and language use. *Proc. International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, Cham, 2016, pp. 28–39.
17. **De Choudhury M.** Role of social media in tackling challenges in mental health. *Proc. the 2nd International Workshop on Socially-Aware Multimedia*, 2013, pp. 49–52.
18. **Al-Mosaiwi M., Johnstone T.** In an absolute state: Elevated use of absolutist words is a marker specific to anxiety, depression, and suicidal ideation. *Clinical Psychological Science*, 2018, vol. 6, no. 4, pp. 529–542.
19. **Nielsen F.** A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *Proc. #MSM2011*, 2011. https://arxiv.org/pdf/1103.2903v1.pdf (дата обращения: 05.08.2021).
20. **Shao J.** Linear model selection by cross-validation. *Journal of the American Statistical Association*, 1993, vol. 88, no. 422, pp. 486–494.

UDC 004.056
doi:10.31799/1684-8853-2022-1-8-18

Determination of the mental state of users of the social network Reddit based on machine learning methods

A. A. Branitskiy[a,b], PhD, Tech., Senior Researcher, orcid.org/0000-0003-3104-0622, branitskiy@comsec.spb.ru
Y. D. Sharma[c], Student, orcid.org/0000-0003-2491-0167
I. V. Kotenko[a,b], Dr. Sc., Tech., Professor, orcid.org/0000-0001-6859-7120
E. V. Fedorchenko[a,b], PhD, Tech., Senior Researcher, orcid.org/0000-0001-6707-9153
A. V. Krasov[b], PhD, Tech., Associate Professor, orcid.org/0000-0002-9076-6055
I. A. Ushakov[b], PhD, Tech., Associate Professor, orcid.org/0000-0002-6988-9261
[a]St. Petersburg Federal Research Center of the RAS, 39, 14 Line, V.O., 199178, Saint-Petersburg, Russian Federation
[b]The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, 22-1, Bolshevikov Pr., 193232, Saint-Petersburg, Russian Federation
[c]Saint-Petersburg Electrotechnical University «LETI», 5, Prof. Popov St., 197376, Saint-Petersburg, Russian Federation

Introduction: Diagnosing mental illness is a complex process that includes conducting dialogue conversations, analyzing the behavior of the subject and passing specialized tests. The successful solution of this problem can be influenced by both the lack of knowledge and experience of the psychologist, and the presence of contradictory or incomplete initial data on the part of the patient. To eliminate this drawback, expert-based or intelligent systems are being developed. **Purpose:** Development of a technique for determining the mental state of social network users. **Results:** Using machine learning methods, a technique has been developed designed to determine the type of a mental state of social network users. The novelty of the proposed technique is in the usage of a two-step text preprocessing procedure and the construction of several sets of features which describe the emotional mood of social network users at the level of the messages published by them. As the initial data, we have used text messages of users of the social network Reddit. There are three stages in the technique: 1) data collection, 2) data preprocessing, 3) post labeling and feature construction. To assess the functioning of a software tool built on the basis of this technique, four indicators were used: accuracy, precision, recall, and F-measure. The best results are demonstrated with a One-vs-Rest ensemble using linear support vector machines as basic solvers. **Practical relevance:** The investigation results can be used in the construction of auxiliary systems that are aimed at supporting decision-making by psychologists in determining mental disorders.

Keywords — machine learning, social network, mental disorders, emotional mood, support vector machine, convolutional neural network.

For citation: Branitskiy A. A., Sharma Y. D., Kotenko I. V., Fedorchenko E. V., Krasov A. V., Ushakov I. A. Determination of the mental state of users of the social network Reddit based on machine learning methods. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 8–18 (In Russian). doi:10.31799/1684-8853-2022-1-8-18

Financial support
The work was supported in part by RFBR project 18-29-22034 mk and budget topic 0073-2019-0002.

References

1. Xue Y., Li Q., Jin L., Feng L., Clifton D. A., Clifford G. D. *Detecting Adolescent Psychological Pressures from Micro-Blog*. In: *Health Information Science*. Y. Zhang, G. Yao, J. He, L. Wang, N. R. Smalheiser, X. Yin Eds. Lecture Notes in Computer Science, 2014. Vol. 8423. Pp. 83–94.
2. Lin H., Jia J., Guo Q., Xue Y., Li Q., Huang J., Cai L., Feng L. User-level psychological stress detection from social media using deep neural network. *Proc. 22nd ACM International Conference on Multimedia*, 2014, pp. 507–516.
3. Park M., Chiyoung C., Meeyoung C. Depressive moods of users portrayed in Twitter. *Proc. HI-KDD*, Beijing, China, ACM, 2012. Available at: https://nyuscholars.nyu.edu/ws/files/134720119/depressive_moods_kdd.pdf (accessed 5 August 2021).
4. De Choudhury M., Gamon M., Counts S., Horvitz E. Predicting depression via social media. *Proc. of the International AAAI Conference on Web and Social Media*, 2013, AAAI, vol. 7, no. 1, pp. 128–137.

5. Tsugawa Sh., Kikuchi Y., Kishino F., Nakajima K., Itoh Y., Ohsaki H. Recognizing depression from twitter activity. *Proc. 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 3187–3196. doi:10.1145/2702123.2702280
6. Schwartz H., Eichstaedt J., Kern M., Park G., Sap M., Stillwell D., Kosinski M., Ungar L. Towards assessing changes in degree of depression through Facebook. *Proc. the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, 2014, pp. 118–125. doi:10.3115/v1/W14-3214
7. Segalin C., Celli F., Polonio L., Kosinski M., Stillwell D., Sebe N., Cristani M., Lepri B. What your Facebook profile picture reveals about your personality? *Proc. the 25th ACM International Conference on Multimedia*, 2017, pp. 460–468. doi:10.1145/3123266.3123331
8. Reece A. G., Christopher M. D. Instagram photos reveal predictive markers of depression. *EPJ Data Science*, 2017, no. 6, pp. 1–12.
9. Gkotsis G., Oellrich A., Velupillai S., Liakata M., Hubbard T. J. P., Dobson R. J. B., Dutta R. Characterisation of mental health conditions in social media using Informed Deep Learning. *Scientific Reports*, 2017, vol. 7, pp. 1–11.
10. Kim J., Lee J., Park E., Han J. A deep learning model for detecting mental illness from user content on social media. *Scientific Reports*, 2020, vol. 10, pp. 1–6. doi:10.1038/s41598-020-68764-y
11. Devlin J., Chang M.-W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. NAACL-HLT*, Minneapolis, Minnesota, 2019, pp. 4171–4186.
12. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. *Roberta: A robustly optimized BERT pretraining approach*. Preprint, 2019. Available at: https://arxiv.org/pdf/1907.11692.pdf (accessed 5 August 2021).
13. Murarka A., Balaji R., Sushma R. *Detection and classification of mental illnesses on social media using RoBERTa*. Preprint, 2020. Available at: https://arxiv.org/pdf/2011.11226.pdf (accessed 5 August 2021).
14. Branitskiy A. A., Doynikova E. V., Kotenko I. V. Use of neural networks for forecasting of the exposure of social network users to destructive impacts. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 1, pp. 24–33 (In Russian). doi:10.31799/1684-8853-2020-1-24-33
15. Baumgartner J., Zannettou S., Keegan B., Squire M., Blackburn J. The Pushshift Reddit dataset. *Proc. the International AAAI Conference on Web and Social Media*, 2020, vol. 14, pp. 830–839.
16. Losada D. E., Crestani F. A test collection for research on depression and language use. *Proc. International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, Cham, 2016, pp. 28–39.
17. De Choudhury M. Role of social media in tackling challenges in mental health. *Proc. the 2nd International Workshop on Socially-Aware Multimedia*, 2013, pp. 49–52.
18. Al-Mosaiwi M., Johnstone T. In an absolute state: Elevated use of absolutist words is a marker specific to anxiety, depression, and suicidal ideation. *Clinical Psychological Science*, 2018, vol. 6, no. 4, pp. 529–542.
19. Nielsen F. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *Proc. #MSM2011*, 2011. Available at: https://arxiv.org/pdf/1103.2903v1.pdf (accessed 5 August 2021).
20. Shao J. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 1993, vol. 88, no. 422, pp. 486–494.

УВАЖАЕМЫЕ АВТОРЫ!

Научные базы данных, включая SCOPUS и Web of Science, обрабатывают данные автоматически. С одной стороны, это ускоряет процесс обработки данных, с другой — различия в транслитерации ФИО, неточные данные о месте работы, области научного знания и т. д. приводят к тому, что в базах оказывается несколько авторских страниц для одного и того же человека. В результате для всех по отдельности считаются индексы цитирования, что снижает рейтинг ученого.

Для идентификации авторов в сетях Thomson Reuters проводит регистрацию с присвоением уникального индекса (ID) для каждого из авторов научных публикаций.

Процедура получения ID бесплатна и очень проста, есть возможность провести регистрацию на 12-ти языках, включая русский (чтобы выбрать язык, кликните на зеленое поле вверху справа на стартовой странице): https://orcid.org

UDC 004.27+004.056
doi:10.31799/1684-8853-2022-1-19-29

Articles

Model of a distributed information system solving tasks with the required probability

V. V. Gryzunov[a], *PhD, Tech., Assistant Professor, orcid.org/0000-0003-4866-217X, viv1313r@mail.ru*
[a]*Russian State Hydrometeorological University, 79, Voronejskaya St., 192007, Saint-Petersburg, Russian Federation*

Introduction: *Distributed in space-time Networks: IIoT and IoT, fog- and edge-computing tend to penetrate into all spheres of human activity. Enterprises, government, law enforcement agencies, etc. depend on the quality of those technologies.* ***Purpose:*** *To determine the composition of the Network that provides the required uptime probability.* ***Methods:*** *According to the concept of structural and functional synthesis, a distributed Network is presented as an unstable queuing system in which servicing devices are connected and disconnected at an arbitrary point in time. A simulation model of the Network has been built.* ***Results:*** *The state of the Network depends on the number of devices and tasks, their performance and lifetimes. The model does not use these quantities themselves, but their ratios. The values of the uptime probability of the Network are calculated for all possible combinations of ratios. The confidence interval has been calculated with a confidence level of 0.95. From the data obtained, it is clear: 1) what should be the minimum composition of the Network in order to provide the required probability; 2) what probability the current composition of the Network can provide; 3) what flow of tasks is admissible in order to solve tasks with the required probability. It is shown that the dependence of the mean tasks residence time on the Network on the composition of the Network has two inflection points. Using information about these points, the Network Management System forms pools of devices or increases the number of devices.* ***Discussion:*** *It is assumed that the Net has a fully connected structure. Consequently, for practical application, it is necessary: to expand the model with an adjacency matrix describing the connections between nodes, and hence the paths of propagation of tasks over the Network or consider that each node is a relay and is capable of transmitting the task to any other node on the Network. Overhead costs arising from this are taken into account by adjusting the original data.* ***Practical relevance:*** *The results allow minimizing costs in the design and operation of distributed systems, maximizing the likelihood of system uptime under given constraints for resource.*

Keywords *— fog computing, edge computing, distributed computing, failure-tolerant computing, QoS, availability.*

For citation: Gryzunov V. V. Model of a distributed information system solving tasks with the required probability. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 19–29. doi:10.31799/1684-8853-2022-1-19-29

Introduction

Distributed computing IoT, IIoT (Industrial Internet of Things), edge, fog and others (hereinafter referred to as Networks) are integrated into the information systems of large enterprises: international meteorological data processing centers, control systems for operational-search activities [1], companies implementing Industry 4.0 standards, etc. On the one hand, the distribution of computations is an inevitable process [2], on the other hand, the welfare of humanity depends on whether the distributed Network can guarantee the solution of the tasks, that is, whether the Network ensures the required quality.

"Quality is a property or a set of properties of an object that determine its compliance with the purpose".

Existing research model the Network differently and evaluate the quality of the Network differently.

At the same time, the Network is characterized by uncertainty associated: 1) with the tasks being solved, 2) with the structure of the Network. This uncertainty has a different nature: 1) stochastic, 2) non-stochastic, i. e. one that cannot be described using models and methods of probability theory, for which all alternatives or a probabilistic description are not known.

Existing studies model the Network differently, and measure the quality of the Network differently and struggle against uncertainties.

Research overview

A number of authors approach this problem by minimizing the time it takes to solve tasks by the Network. Only the uncertainty of the tasks to be solved is touched upon.

For example, researchers [3] reduce the time by using a special algorithm for randomly sending tasks between nodes. The decision on whether to take a task to work or send it to another node is taken independently by the Network node. It is assumed that if a node has accepted a task for execution, then it will necessarily execute it or send it to the one who will execute it. Cases with sudden failures or disconnection of nodes from the Network are ignored.

The authors [4] use the delay time of the Network's response to a service request by the user. It is proposed to reduce latency by replicating da-

ta in a distributed geographic information system. If the Network is large, then the iFogStorP method must be used, otherwise the iFogStor. It is believed that the probability of solving tasks by the Network is 100%.

Researchers in [5] went further. They have developed mathematical models of the Network as a queuing system (QS) (Multiple Input Multiple Output) and propose to reduce the time to receive a service by changing the discipline of service. According to the results of the study, it can be seen that the minimum residence time on the Network is achieved when using a strategy that chooses to serve a subset of users with the maximum bandwidth. And in some cases, a strategy that organizes services in the order in which tasks are received. Denial of service is provided only if the queue does not reach the task.

The next group of works is focused on eliminating the structure uncertainty of a stochastic nature. The quality of service of the Network estimate through the fault tolerance of the Network. Fault tolerance is increased by introducing redundancy in the form of backups or clustering.

Research [6] concerns fault-tolerant energy-efficient routing in a distributed system based on IoT sensors. Clusters are created on the basis of two groups of nodes: normal and powerful. The powerful nodes control the load and activity of the normal nodes. The number of Network nodes is determined at the initial moment of time and does not increase during operation. The duration of solving tasks is ignored. The study concludes: the higher the network fault tolerance is required, the more nodes are needed; the higher the probability of node failure, the more nodes are needed to provide the required fault tolerance. However, the study has no data on how many nodes are needed.

On the other hand, the more nodes, the more overhead, the more network management overhead, the more difficult it is to balance the load. A similar problem was discovered by the authors of [7].

Overhead costs and load balancing to improve fault tolerance by optimizing the probability of failure-free operation are considered by the authors in [8]. The authors conclude that for an information and control system processing distributed data, a distributed structure is better suited (in terms of the uptime probability). At the same time, the authors' model considers the "fog layer" as a single device and takes into account the possibility of its failure only due to overheating caused by a large computational load. The possibility of restoring or changing the probabilistic characteristics of the "fog layer" during the operation period is not taken into account by the model.

The work [9] is devoted to the study of the dependence of the uptime probability on the relative speed of node recovery. The network is represented as a QS described by the Kolmogorov — Chapman system of differential equations. The dependences of the uptime probability of the system on the relative speed of recovery are obtained for various distributions of the repair time of nodes (Weibull — Gnedenko, Pareto and Lognormalnoe). However, the authors impose a rather strong restriction — they require that the system have a stochastic nature and, moreover, have a stationary probability distribution. In [10], the indicated restrictions are toughened by the requirement of the ordinariness of the flow of requests and the flow of refusals of nodes. Fault tolerance is a function of the network's mean response time to user requests.

The next line of works is exploring the possibility of improving the quality of service through compute scheduling, load balancing and routing.

The authors of [11] propose decentralized load balancing of heterogeneous nodes of the Network to increase the probability of solving tasks. Limitations adopted in the work: the structure of the Network is constant, the nodes solve the accepted tasks with a probability of 100%. Only the uncertainty of the tasks being solved is taken into account.

In [12], the load is controlled centrally based on templates. Templates are created in advance. Small variability of the structure of the Network is possible within the given templates. Tasks and nodes are classified according to their computational nature. Nodes accept tasks of their class.

In the articles studying the behavior of Mobile-edge Computing networks, some restrictions on the fixed structure of the Network are removed, the combinatorial complexity of load planning and the heterogeneity of nodes are taken into account [13]. In [14], researchers propose intelligent planning.

At the same time, the authors imply that the performance of devices and their number are predetermined, therefore, their research may be applicable only to some fairly stationary segments of the Network, such as clouds, but not fog, and even more so edge computing. The structure and functions of the Network, using edge computing, IoT, IIoT, are changing so much that the authors of the study [15] are forced to impose a ban on leaving node the Network until the node solves the task.

As can be seen from the analysis, the existing works operate with models that are not entirely adequate for the Network, do not fully take into account the uncertainty that is characteristic of the Network, namely: resource intensity of tasks; the ability of the nodes of the Network to arbitrarily connect and disconnect from the Network; options when a node takes a task, does not perform it, because it leaves the Network; situations with denial of service, when there are free nodes in the Network,

but they refuse tasks, because their performance is not enough to solve the task.

A more adequate model of the Network will make it possible to more accurately predict the behavior of the Network in conditions of uncertainty and reasonably put forward requirements for its composition. To solve the indicated problem of modeling a distributed Network, solving the tasks with the required probability under conditions of strong uncertainty, it is necessary to formulate it, choose the appropriate indicator of the efficiency of the Network's functioning and methods of its calculation.

Strong uncertainty is understood as uncertainty associated with the tasks being solved and the structure of the Network and having a stochastic or non-stochastic nature.

"Efficiency is a property (quality) of the system functioning process, defined as its adaptability to solving the tasks set for the system" [16, p. 31].

Statement of the research problem

The formulation of the problem is based on the concept of structural and functional synthesis of systems, presented in [17] and the keep integrity law of an object, which states that there is a stable repeating connection between the properties of an object and its actions with a fixed purpose of the object:

$$I(Q) = F(Q, \Phi(R, U), t), \quad (1)$$

where:

I is an indicator of the effectiveness of the Network, the presentation of the required number of required characters at the required time (the number of simultaneously tracked targets, the number of web-portal users, the probability of image recognition, etc.);

Q — the set of required space-time states of the Network (Network model) is set by the metasystem / Network creator. Shows how exactly the elements of the Network should interact with each other and with the Network users. In the general case, it is a function of time and is set in various ways: by enumeration, analytically, by specifying characteristic properties, etc.;

Φ — the set of current space-time states of the Network, is a model of the current situation, shows how the elements of the Network interact with each other and the user of the Network in reality;

R is the set of capacities of the elements of the Network (model of actions of the elements of the system in space-time);

U is a set of control actions on the elements of the Network;

t is the Network operation time;

F is an operator expressing the basic laws of the existence of the Network.

All types of uncertainties of any nature inherent in an object are manifested through Q, R and Φ and are taken into account in expression (1).

It is advisable to begin the specification of expression (1) describing the Network with the formalization of the performance indicator (I).

Indicator of the effectiveness of the Network as a system operating in conditions of strong uncertainty

Systems with high uncertainty have their own performance indicators and methods of calculating them. The advantages and limitations of some indicators are analyzed in [18]. This study uses an indicator borrowed from [18] — the uptime probability of Network (UPN):

$$P = K^*/K, \quad (2)$$

where P — UPN; the indicator can be called a probability because it satisfies the corresponding axiomatics; K, K^* — the number of tasks assigned to the Network and performed by the Network respectively for the entire period of operation.

The indicator (P) is integral and uses only the number of tasks set and solved. The number of tasks set and their characteristics depend on the user's goals, the number of tasks solved depends on the characteristics of the tasks themselves, the available resources of the Network, their properties and configuration. Consequently, the indicator takes into account the uncertainty of any nature associated with both the tasks and the structure of the Network. And it can be called a probability, because it satisfies the corresponding axiomatics.

Axiom 1. The event consists in the solution of the i-th task by the Network. The solution by the Network of all assigned (K) tasks forms a complete group of events $\mathcal{F}$. An arbitrary system of subsets of the set $\mathcal{F}$ is closed under the operations of complement and union, and, therefore, is an algebra of events.

Axiom 2. Each i-th event, consisting in solving the i-th task, is associated with a non-negative real number K_i/K, where K_i is the same for all tasks and is equal to 1, therefore, each individual event is associated with the number $1/K$.

Axiom 3. $P(0) = 0/K = 0$, $P(K) = K/K = 1$, therefore $0 \le P \le 1$.

Axiom 4. For disjoint events $i \ne j$, $P(K_i) + P(K_j) = P(K_i + K_j)$.

It is not very convenient to calculate UPN in this way for a real Network, because it can be calculated

only after the Network stops functioning. However, it is well applicable for studying the properties of the Network on the model.

In this case, the research problem is set as follows.

Verbal problem setting

Let's represent the Network as a set of QS $G/G/1$, that is, the laws of claims arrival and their servicing are arbitrary. The queue is endless. It is necessary to find the UPN of Network.

At first glance, the problem of specifying type of operator (F) in expression (1) looks like an ordinary problem of the theory of queuing, solved analytically. One of the typical works that solve such a problem [19] considers the Network as a set of QS $M/M/2$ with one reliable and one unreliable element, takes into account the variability of the Network structure and even delays in the dissemination of information on the Network about a failed device.

However, upon a more detailed study, it becomes clear that some details of real Networks cast doubt on the advisability of searching for expression (1) solutions in an analytical form. The most typical of these details:

— in a real Network, different tasks have different needs for the performance of service devices, including the minimum permissible performance of the processing device. In the QS, this is solved either by classifying claims, or by specifying a service flow that takes into account both the variability of the device in the form of available performance and various requirements of claims. Both are challenging in themselves;

— in the course of life on the Network, situations arise that are of a purposeful aggressive nature that cannot be described in terms of probability theory and mathematical statistics, for example, cyberattacks;

— nodes join the Network and leave it at an arbitrary moment in time;

— a node starts to solve a task and does not finish;

— a task arrives in the system through an arbitrary node of the Network.

There are ways to take into account the indicated features and obtain a solution in an analytical form: classification of claims and service devices, phase method, methods of working with devices with variable structure, etc. But the analytical solution is usually private with strong restrictions and assumptions, and the final system of equations is solved simply for 20–30 devices. And usually only numerically. The number of nodes in a real network can be hundreds of thousands of nodes.

Considering the above, it seems appropriate to solve the problem by drawing up a simulation model of the Network. For this, it is necessary to take into account the information and control connections of the tasks being solved.

Accounting for the information and control connections between tasks

As a rule, the tasks performed are linked by information and control links. And it can be assumed that the likelihood of completing one depends on the likelihood of completing the related task. Taking these dependencies into account greatly complicates the Network model. In the study [18], it was shown that for calculating the UPN, information and control dependencies can be neglected and the UPN can be calculated separately for each task:

"*Theorem*. Let a directed graph G be given, reflecting the dependencies between tasks. Then the events involved in solving tasks are independent for any directed graph.

Supposition. The considered technical systems are dynamic, therefore, the situation at the current time does not affect the situation at previous times (the situation at subsequent times does not affect the situation at the current times)".

Formal statement of the research problem

Performance is the number of tasks solved per unit of time.

Given:

1) T — Network lifetime (*MaxModelTime* in the model);

2) P^* — required UPN;

3) P — current UPN;

4) maximum available node performance (Ω_{node}) — describes the node's ability to provide its resources to the task;

5) the minimum performance required by the task (Ω_{task}) — the lower estimate of the node performance. If the performance of the node is less than the minimum required, then the node does not take the task, even if it is free: $\Omega_{node} \geq \Omega_{task}$;

6) maximum task execution time (T_{task}) — the time that cannot be exceeded by the node if the node has taken the task to work;

7) maximum lifetime of nodes (T_{node}) — describes the degradation of the Network, a node disappears from the Network if its residence time is greater than the maximum. If he solves the task, the task is considered lost;

8) maximum number of incoming tasks at each moment of time (A_{task}) — characterizes the load of the Network at each moment of time;

9) maximum number of nodes connecting to the Network at each moment of time (A_{node}) — describes

the increase in the resources of the Network at each moment of time.

Required:

get such a Network configuration for which the required UPN will be less than or equal to the current one:

$$P^* \leq P.$$

The simulation model was compiled and investigated in the MatLab language.

Network simulation model

Initial data

From the condition of the problem described above, it follows that the configuration of the Network is determined by nine values.

To simplify modeling, the model uses not the quantities themselves, but their ratios:

1) $CurrentTaskDuration = T_{task}/T_{node}$;
2) $CurrentTaskPerfomance = \Omega_{task}/\Omega_{node}$;
3) $CurrentRatio = A_{task}/A_{node}$.

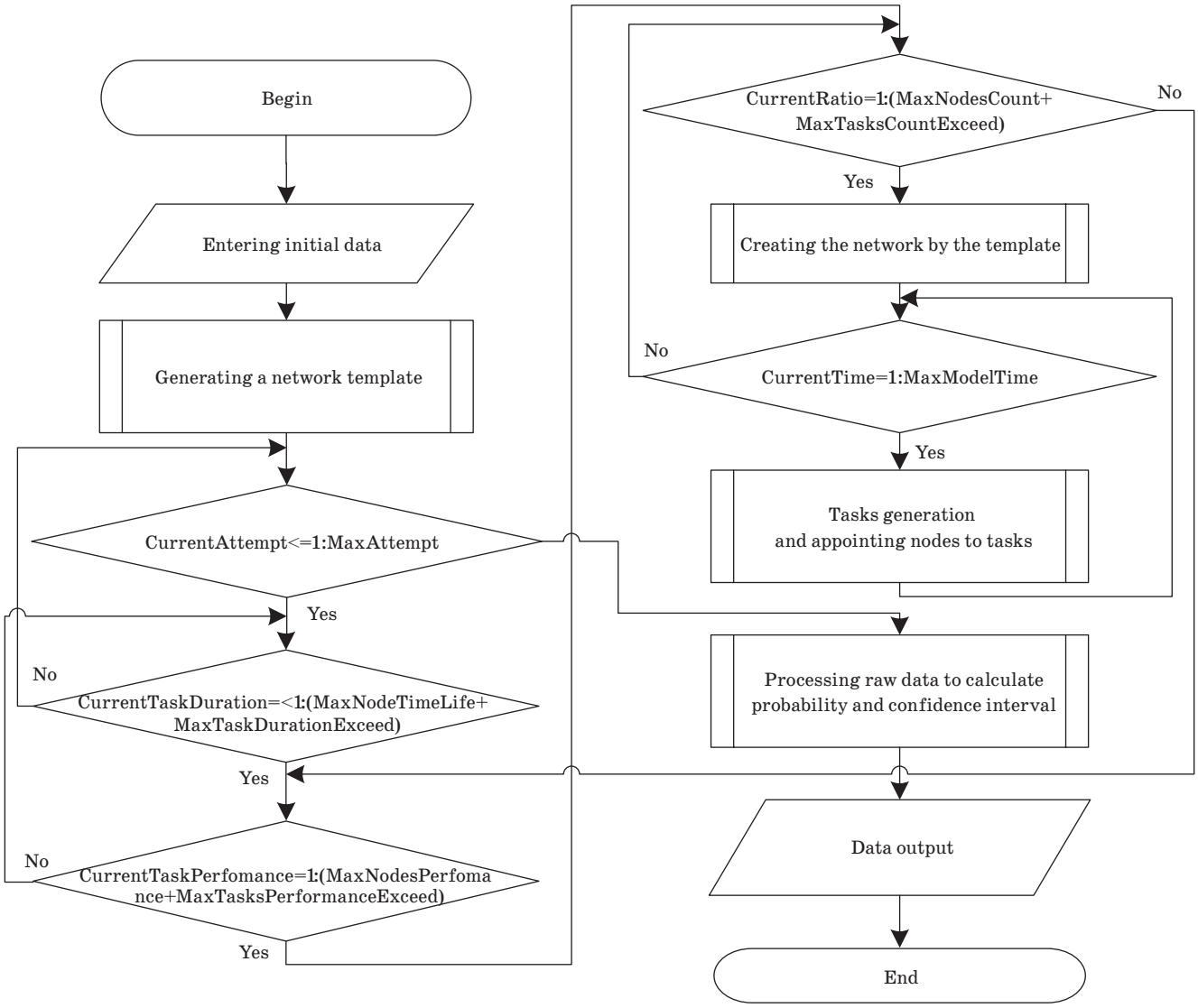
Figure 1 has a simplified block diagram of the simulation model algorithm.

It can be seen from the given algorithm that the interaction of the same Network configuration with different task flows is simulated.

Network configurations are generated randomly according to a uniform distribution law with the following parameters:

$T_{node} \in [1; MaxNodeTimeLife]$;
$\Omega_{node} \in [1; MaxNodesPerfomance]$;
$A_{node} \in [1; MaxNodesCount]$.

The Network configuration is invariant. Relative to it, other values of the model, generated randomly according to the uniform distribution law, change:



■ ***Fig. 1.*** Simplified block diagram of a simulation model

$T_{task} \in [1; (MaxNodeTimeLife+MaxTaskDuration\ Exceed)]$;

$\Omega_{task} \in [1; (MaxNodesPerfomance+MaxTasks\ PerformanceExceed)]$;

$A_{task} \in [1; (MaxNodesCount+MaxTasksCount\ Exceed)]$.

The required UPN (P^*) is also constant for all variations. The network lifetime (T) is set by the simulation time, is stored in the *MaxModelTime* variable and takes on a significantly larger value than other times in the model. This improves confidence in the simulation results.

Tasks appear and are executed at an arbitrary moment of the Network functioning, there are no deadlines, therefore, the resulting UPN is the upper estimate of the real UPN.

The task is lost if the node accepts the task and leaves the Network without completing the solution.

Of all the available nodes for solving the task, the node is selected whose maximum available performance is closest to the minimum performance required by the task. The model implements the FIFO (first input first output) algorithm, i. e. the task that entered the Network earlier is assigned first. Among the devices, the one that connected to the Network earlier is selected.

Real nodes of the Network have physical restrictions on the number of connections with other nodes, therefore, in the model, the value describing the number of nodes is random and bounded from above.

The minimum performance required by the i-th task ($\omega_{i,\ task} \in (0; \Omega_{task}]$), the time to solve the task when this productivity is obtained $t_{i,\ task} \in (0; T_{task}]$ and the maximum available performance of the j-th node ($\omega_{j,\ node} \in (0; \Omega_{node}]$) determine the time during which the node is considered busy and cannot accept other tasks($t_{j,\ busy}$):

$$t_{j,\ busy} = t_{i,\ task}\omega_{i,\ task}/\omega_{j,\ node}.$$

Simulation results

Due to the large amount of data received, in Tables 1 and 2, only a part of them is shown for *CurrentTaskDuration* = 6/10. The rest of the data, if necessary, can be provided upon request. The mean UPN values for all possible Network configuration options are given in the Table 1. The confidence interval with a confidence level of 0.95 increases with the growth of the *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio* ratios and lies in the interval [0.0013; 0.056]. An increase in the confidence interval is associated with an increase in the intervals in which the simulated values are located:

$T_{task} \in [1; (MaxNodeTimeLife+MaxTaskDuration\ Exceed)]$;

$\Omega_{task} \in [1; (MaxNodesPerfomance+MaxTasks\ PerformanceExceed)]$;

$A_{task} \in [1; (MaxNodesCount+MaxTasksCount\ Exceed)]$.

It can be seen from the figure that the smaller the task/node ratio, the higher the UPN. At the same time, the unit costs for one task increase [7]. However, the UPN is influenced not only by this ratio, but also by the nature of the tasks performed, therefore, the Network Management System can provide the required UPN, based on the duration of incoming tasks and their performance requirements.

The mean tasks residence time in the Network for *CurrentTimeRatio* = 6/10 is presented in the

■ ***Table 1***. Mean UPN for all experiments performed, *CurrentTimeRatio* = 6/10

CurrentRatio	CurrentTaskPerfomance										
	1	2	3	4	5	6	7	8	9	10	11
1	0.717	0.677	0.622	0.599	0.566	0.520	0.517	0.524	0.548	0.476	0.437
2	0.771	0.704	0.672	0.623	0.598	0.583	0.554	0.562	0.547	0.487	0.465
3	0.786	0.727	0.695	0.671	0.629	0.603	0.582	0.593	0.577	0.526	0.466
4	0.821	0.760	0.709	0.688	0.669	0.618	0.638	0.607	0.611	0.527	0.491
5	0.828	0.773	0.721	0.688	0.662	0.662	0.630	0.625	0.624	0.550	0.480
6	0.843	0.777	0.748	0.709	0.682	0.668	0.654	0.646	0.631	0.569	0.516
7	0.835	0.794	0.757	0.728	0.697	0.674	0.660	0.648	0.651	0.580	0.507
8	0.859	0.799	0.756	0.717	0.716	0.690	0.680	0.669	0.656	0.602	0.527
9	0.864	0.805	0.777	0.753	0.718	0.702	0.692	0.686	0.675	0.595	0.536
10	0.869	0.822	0.782	0.764	0.728	0.717	0.707	0.693	0.672	0.605	0.537
11	0.868	0.826	0.793	0.751	0.737	0.720	0.709	0.696	0.684	0.606	0.544

Table 2. Times are calculated only for those tasks that were performed by the Network. Lost tasks or tasks that were not recruited for any reason were not counted.

The simulated Network was tested 30 times. The duration of the Network functioning is 100 units of model time. Fig. 2 contains a surface, inside which all configurations of the Network are located, which solves the assigned tasks with an UPN of at least 0.8.

Consequently, this simulation result can be interpreted as follows: the tasks entering the Network are solved with probability (see the Table 1). For tasks that are solved by the Network, they are on the Network on mean (see the Table 2) units of model time. Tasks are denied service for the following reasons:

1) the minimum required performance for the task is greater than the current performance of any node on the Network;

2) the node leaves the Network, taking the task and not finishing its solution. The reason for leaving is not important for the simulation results — it can be a failure of the software or hardware of the node, disconnection of the node from the Network, disabling the node during a cyber attack, etc.

Figure 3 shows the dependence of the mean tasks time on the Network on various Network configurations. The graph was built as follows:

1) each *CurrentTaskDuration* ratio has its own legend (line of the same color);

2) found the mean value for each row from the Table 2 (*MeanTimeForCurrentRatio*). This value shows the mean tasks residence time on the Network with a fixed *CurrentTaskPerfomance*;

3) found the mean value for each column from the Table 2 (*MeanTimeForCurrentTaskPerfomance*). This value shows the tasks residence time on the Network with a fixed *CurrentRatio*;

4) the difference between the values obtained in steps 2 and 3 was calculated: *MeanTimeFor CurrentRatio–CurrentTaskPerfomance* for each row-column pair;

5) a graph of the difference was built for each number of the pair from step 4.

The graph in Fig. 3 shows a pattern: there are clearly pronounced inflection points numbered 6 and 9. These points correspond to the 6/10 and 9/10 ratios for the *CurrentRatio* and *CurrentTaskPerfomance* ratios. It is interesting that through the same points the surface corresponding to UPN 0.8 (see Fig. 2) cuts the plane formed by the lines *CurrentRatio* and *CurrentTaskPerfomance*.

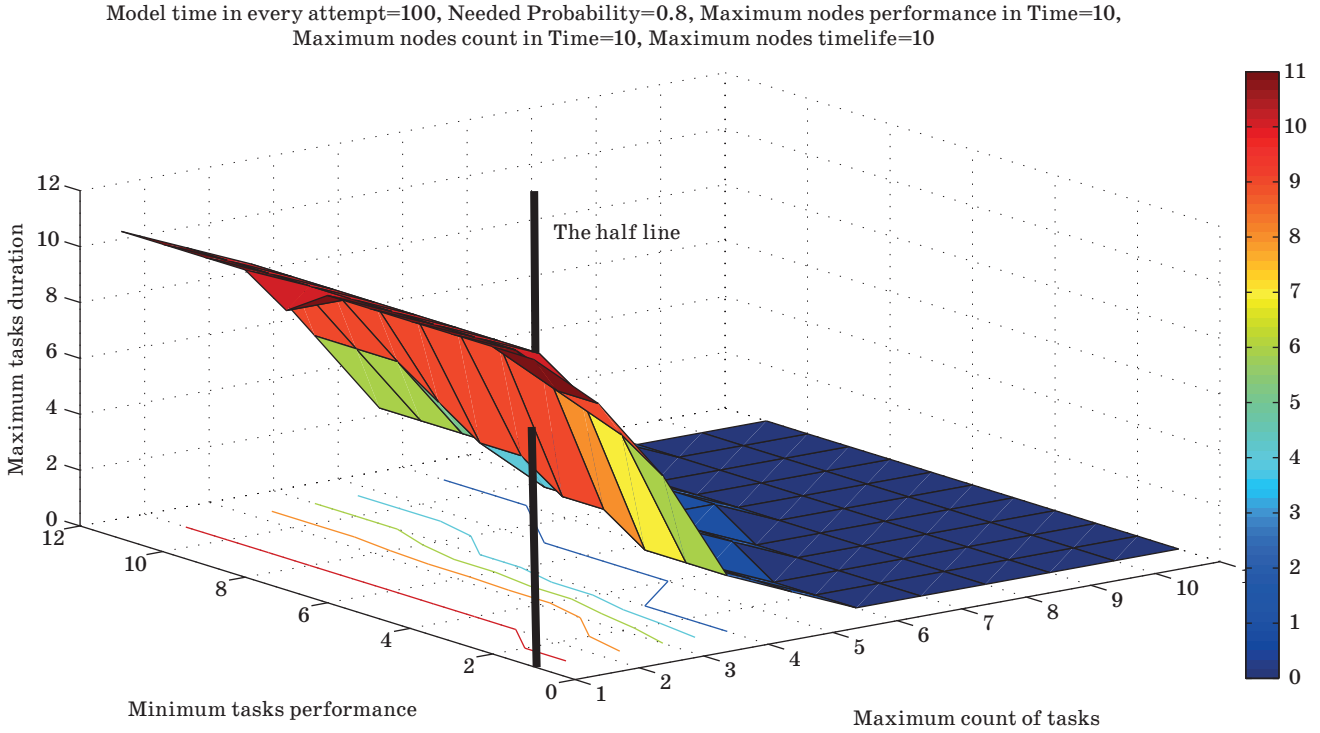
Figure 3 shows two distinct situations:

1. Graph in the positive half-plane. This means that with an increase in the number of tasks, the mean tasks residence time on the Network grows faster than with an increase in the minimum required performance. This happens as long as the *CurrentRatio* < 6/10.

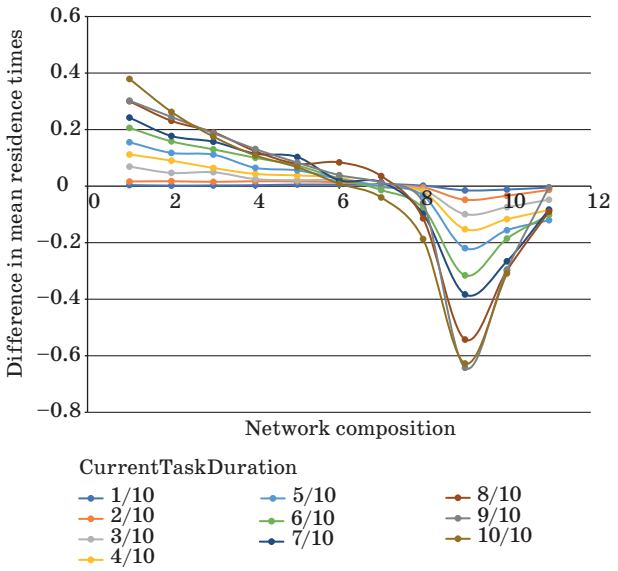
2. Graph in the negative half-plane. This means that with an increase in the number of tasks, the

■ ***Table 2***. Mean residence time on the Network of tasks solved by the Network, *CurrentTimeRatio* = 6/10

The set number	The set number											*Mean Time For Current TaskPerfomance*
	1	2	3	4	5	6	7	8	9	10	11	
1	0.708	0.760	0.824	0.852	0.928	0.939	1.007	1.082	1.188	1.183	1.253	**0.97**
2	0.729	0.782	0.835	0.911	0.947	0.970	1.056	1.084	1.248	1.369	1.162	**1.01**
3	0.741	0.800	0.865	0.926	0.978	1.081	1.086	1.175	1.417	1.243	1.387	**1.06**
4	0.739	0.821	0.885	0.973	1.017	1.110	1.146	1.240	1.500	1.424	1.348	**1.11**
5	0.753	0.836	0.919	1.013	1.063	1.126	1.196	1.304	1.558	1.562	1.470	**1.16**
6	0.755	0.847	0.950	1.019	1.102	1.195	1.232	1.357	1.635	1.536	1.422	**1.19**
7	0.767	0.867	0.960	1.034	1.131	1.164	1.282	1.371	1.708	1.575	1.525	**1.22**
8	0.812	0.892	0.963	1.058	1.171	1.230	1.309	1.430	1.665	1.758	1.644	**1.27**
9	0.815	0.902	0.981	1.090	1.189	1.258	1.370	1.479	1.890	1.711	1.791	**1.32**
10	0.811	0.899	1.047	1.109	1.211	1.332	1.399	1.569	2.070	2.033	1.749	**1.38**
11	0.825	0.946	1.041	1.116	1.261	1.351	1.452	1.698	2.066	1.878	1.822	**1.41**
Mean Time For Current-Ratio	**0.77**	**0.85**	**0.93**	**1.01**	**1.09**	**1.16**	**1.23**	**1.34**	**1.63**	**1.57**	**1.51**	*Mean Time For CurrentRatio*
Difference in mean residence times	**0.21**	**0.16**	**0.13**	**0.10**	**0.07**	**0.03**	**−0.01**	**−0.08**	**−0.32**	**−0.19**	**−0.10**	*Difference in mean residence times*



■ ***Fig. 2.*** Network configuration solving the assigned tasks with a UPN of at least 0.8



■ ***Fig. 3.*** Dependence of the mean tasks residence time on the Network on various Network configurations

mean tasks residence time on the Network grows more slowly than with an increase in the minimum required performance. This happens when *CurrentRatio* > 7/10.

Hence it follows that with an increase in the load on the Network and *CurrentRatio* < 6/10, it is better to first of all create pools [20] from the existing nodes. Better in the sense of reducing the mean residence time on the Network while maintaining the required UPN. If the *CurrentRatio* > 7/10, it is better to focus on connecting new nodes.

In [21], a method is described, which can be used to use the time and accuracy errors admissible for tasks and to weaken the requirements for the Network, while maintaining the required UPN. In Fig. 2, this appears as a shift of the surface to the right along the ray outgoing from the origin. The beam tilt angles depend on the tolerances for the task.

The work [22] describes an example of patrolling of a naval base by unmanned aerial vehicles "Orlan". As a result of information and technical impact, a situation arises when the available resource of the unmanned aerial vehicles is not enough to solve the assigned task of monitoring the perimeter, and the control system forms a pool with the required performance. A similar situation of resource scarcity and a decrease in UPN occurs with an increase in the number of tasks or their resource intensity. Using the data from the proposed study, we can say that in order to achieve the required UPN, it is necessary to keep the ratios *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio* in the specified range. This is achieved either by forming resource pools or by using task reserves in terms of time and accuracy [21].

Directions for further research

The developed model assumes that the Network has a full-mesh topology. In the future, it is supposed to use the adjacency matrix and study the behavior of Networks with different topologies, compare the work of the method offered in [20] with others.

It is considered that the node that accepts the task solves it with 100% probability. The reliability of the nodes themselves can be taken into account either by creating a special module that thins the flow of nodes according to the required pattern, or by making changes to the initial data, on the basis of which the Network model is generated.

Events consisting in the appearance of tasks and the connection of nodes to the Network and disconnection of nodes from it are random in the presented model. This is well suited to describe normal operation: solving planned tasks, failures and failures caused by natural causes. But this does not fully reflect the targeted aggressive effects on the Network: cyberattacks, information technology intrusions, a surge in tasks in case of emergencies, etc.

Conclusion

With the help of the approach proposed in the article, a significant step has been taken in concretizing the expression for the structural and functional synthesis of the Network (1), namely: such characteristics of a distributed Network as the UPN and the average time spent on the Network were formulated and studied through the simulation model; the variables on which the state of the Network depends were indicated. The number of variables has been reduced almost twice by introducing the ratios *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio*; shown, that with further concretization of expression (1) Information-control dependencies between tasks can be neglected.

Each required UPN of Network is matched its own surface. To ensure the solution of task with the required UPN, the Network control system needs to keep the values of the *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio* ratios under this surface.

The graph describing the behavior of the mean residence time of tasks on the Network has two distinct inflection points. Knowledge of these points helps the control system to make a decision on the choice of an acceptable ratio of *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio*.

By simple modifications of the model, it is possible to study the behavior of the Network under conditions of non-stochastic uncertainty, which cannot be described in terms of probability theory: cyberattacks, information technology intrusions, a surge in tasks in case of emergencies, etc.

Financial support

The reported study was funded by Russian Ministry of Science (information security), project No 08/2020.

References

1. Kudelkin V. A., Denisov V. F. Experience of integration of distributed information systems. *IT-standart*, 2017, no. 1. Available at: http://journal.tc22.ru/wp-content/uploads/2018/02/opit_integracii_raspredelennih_informacionnih_sistem.pdf (accessed 03 September 2021) (In Russian).
2. Abdulqadir H. R., Zeebaree S. R., Shukur H. M., Sadeeq M. M., Salim B. W., Salih A. A., Kak S. F. et al. A study of moving from cloud computing to fog computing. *Qubahan Academic Journal*, 2021, vol. 1, no. 2, pp. 60–70. doi:10.48161/qaj.v1n2a49
3. Yousefpour A., Ishigaki G., Jue J. P. Fog Computing: Towards minimizing delay in the Internet of Things. *Proceedings — 2017 IEEE 1st International Conference on Edge Computing*, Edge, 2017, pp. 17–24. doi:10.1109/IEEE.Edge.2017.12
4. Naas M. I., Lemarchand L., Raipin P., Boukhobza J. IoT data replication and consistency management in fog computing. *Journal of Grid Computing*, 2021, vol. 19, iss. 33. doi:10.1007/s10723-021-09571-1
5. Gorbunova A. V., Medvedeva E. G., Gaidamaka Yu. V., Shorgin V. S., Samouylov K. E. Effective user service strategies in a multi-user MIMO system. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 4, pp. 69–81 (In Russian). doi:10.31799/1684-8853-2019-4-69-81
6. Lin J., Chelliah P. R., Hsu M., Hou J. Efficient fault-tolerant routing in IoT wireless sensor networks based on bipartite-flow graph modeling. *IEEE Access*, 2019, vol. 7, pp. 14022–14034. doi:10.1109/ACCESS.2019.2894002
7. Grover J., Garimella R. M. Reliable and fault-tolerant IoT-edge architecture. *IEEE Sensors*, IEEE, 2018, pp. 1–4. doi:10.1109/ICSENS.2018.8589624
8. Melnik E. V., Klimenko A. B., Ivanov D. Y. Distributed information and control system reliability enhancement by fog-computing concept application. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2018, vol. 327, no. 2, pp. 022070. doi:10.1088/1757-899X/327/2/022070
9. Kinmanon U. G. Zh., Kozyrev D. V. Analytical and simulation modeling of the reliability of a closed ho-

mogeneous system with an arbitrary number of data sources and limited resources for their processing. *Modern Information Technology and IT-education*, 2018, vol. 14, no. 3, pp. 552–559 (In Russian). doi:10.25559/SITITO.14.201803.552-559

10. Skoba A. N., Mikhaylov V. K., Panfilov A. N. The problem of determining the optimal fault tolerance of distributed information processing systems. *Engineering Journal of Don*, 2020, no. 4. Available at: http://www.ivdon.ru/uploads/article/pdf/IVD_28__3_scoba_mikhaylov_ayesh_loganchuk.pdf_fe4c698775.pdf (accessed 06 September 2021) (In Russian).
11. Beraldi R., Canali C., Lancellotti R., Mattia G. P. Distributed load balancing for heterogeneous fog computing infrastructures in smart cities. *Pervasive and Mobile Computing*, 2020, vol. 67, no. 6, pp. 101221. doi:10.1016/j.pmcj.2020.101221
12. Kapsalis A., Kasnesis P., Venieris I. S., Kaklamani D. I., Patrikakis C. Z. A cooperative fog approach for effective workload balancing. *IEEE Cloud Computing*, 2017, vol. 4, no. 2, pp. 36–45. doi:10.1109/MCC.2017.25
13. Pham Q.-V., LeAnh T., Tran N. H., Seon Hong C. Decentralized computation offloading and resource allocation in heterogeneous networks with mobile edge computing. *IEEE Transactions on Mobile Computing*, March 2018. doi:arXiv:1803.00683
14. Peng Q., Wu C., Xia Y., Ma Y., Wang X., Jiang N. DoSRA: A decentralized approach to online edge task scheduling and resource allocation. *IEEE Internet of Things Journal*. doi:10.1109/JIOT.2021.3107431
15. Jia B., Hu H., Zeng Y., Xu T. Double-matching resource allocation strategy in fog computing networks based on cost efficiency. *Journal of Communications and Networks*, 2018, vol. 20, no. 3, pp. 237–246. doi:10.1109/JCN.2018.000036
16. *Elementy teorii testirovaniya i upravleniya tekhnicheskimi sistemami* [Elements of the theory of testing and control of technical systems]. R. M. Yusupov Ed. Leningrad, Energetika Publ., 1978. 191 p. (In Russian).
17. Burlov V. G., Gryzunov V. V., Tatarnikova T. M. Threats of information security in the application of GIS in the interests of the digital economy. *Journal of Physics: Conference Series*, 2020, vol. 1703, pp. 012023 doi:10.1088/1742-6596/1703/1/012023
18. Burlov V. G., Gryzunov V. V. Evaluation of the effectiveness of geographic information systems adaptation to destabilizing factors. *Journal of Physics: Conference Series,* 2020, vol. 1703, pp. 012016. doi:10.1088/1742-6596/1703/1/012016
19. Tananko I. E., Fokina N. P. Method of analysis of queuing networks with unreliable devices and information delay. *Vestnik Tomskogo gosudarstvennogo universiteta. Menedzhment, komp'yuternyye tekhnologii i informatika*, 2020, no. 52, pp. 90–96 (In Russian). doi:10.17223/19988605/52/11
20. Gryzunov V. V. Dynamic aggregation of pools in military computing systems. *Informatsionno-upravlyayushchiye sistemy* [Information and Control Systems], 2015, no. 1, pp. 13–20 (In Russian). doi:10.15217/issn1684-8853.2015.1.13
21. Gryzunov V. V. Problem solving method of measuring and calculating tasks under conditions of data computing system degradation. *Vestnik SibGUTI*, 2015, no. 1 (29), pp. 35–46 (In Russian).
22. Gryzunov V. V. FIST geoinformation system model using fog computing in destabilization. *Herald of Dagestan State Technical University. Technical Sciences*, 2021, no. 48 (1), pp. 76–89 (In Russian). https://doi.org/10.21822/2073-6185-2021-48-1-76-89

УДК 004.27+004.056
doi:10.31799/1684-8853-2022-1-19-29

Модель распределенной информационной системы, решающей задачи с требуемой вероятностью

В. В. Грызунов[a], канд. техн. наук, доцент, orcid.org/0000-0003-4866-217X, viv1313r@mail.ru
[a]Российский государственный гидрометеорологический университет, Воронежская ул., 79, Санкт-Петербург, 192007, РФ

Введение: распределенные в пространстве-времени сети IIoT и IoT, fog- и edge-вычисления имеют тренд проникать во все сферы человеческой деятельности. От качества работы этих технологий зависят предприятия, органы власти, силовые структуры и т. д. **Цель:** определить состав сети, обеспечивающий требуемую вероятность безотказной работы. **Методы:** согласно концепции структурно-функционального синтеза распределенная сеть представлена как нестабильная система массового обслуживания, в которой обслуживающие устройства подключаются и отключаются в произвольный момент времени. Построена имитационная модель сети. **Результаты:** состояние сети зависит от количества устройств и задач, их производительностей и времен жизни. В модели рассматриваются не сами эти величины, а их соотношения. Рассчитаны значения вероятности безотказной работы сети для всех возможных сочетаний соотношений. Рассчитан доверительный интервал с уровнем доверия 0,95. Из полученных данных понятно: 1) каким должен быть минимальный состав сети, чтобы обеспечить требуемую вероятность; 2) какую вероятность может обеспечить текущий состав сети; 3) какой поток задач допустим, чтобы решать задачи с требуемой вероятностью. Показано, что зависимость среднего времени нахождения задачи в сети от состава сети имеет две точки перегиба. Используя сведения об этих точках, система управления сетью формирует пулы из устройств или увеличивает количество устройств. **Обсуждение:** подразумевается, что сеть имеет полносвязную структуру. Следовательно, для практического применения необходимо расширить модель матрицей смежности, описывающей связи между узлами, а значит, пути распространения задач по сети, или считать, что каждый узел является ретранслятором и способен передать задачу на любой другой узел сети. Накладные издержки, возникающие при

этом, учитываются через корректировку исходных данных. **Практическая значимость:** результаты позволяют минимизировать издержки при проектировании и эксплуатации распределенных систем, максимизировать вероятность безотказной работы систем при заданных ограничениях на ресурсы.

Ключевые слова — туманные вычисления, edge-вычисления, распределенные вычисления, отказоустойчивые вычисления, QoS, доступность.

Для цитирования: Gryzunov V. V. Model of a distributed information system solving tasks with the required probability. *Информационно-управляющие системы*, 2022, № 1, с. 19–29. doi:10.31799/1684-8853-2022-1-19-29

For citation: Gryzunov V. V. Model of a distributed information system solving tasks with the required probability. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 19–29. doi:10.31799/1684-8853-2022-1-19-29

Финансовая поддержка

Исследование выполнено при финансовой поддержке Министерства науки России (информационная безопасность), проект № 08/2020.

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (ius.spb@gmail.com).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию.

Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.

UDC 004.05
doi:10.31799/1684-8853-2022-1-30-43

Articles

Kex: A platform for analysis of JVM programs

A. M. Abdullin[a,b], *Post-Graduate Student, Assistant Professor, orcid.org/0000-0002-9669-2587*
V. M. Itsykson[a,b], *PhD, Tech., Associate Professor, orcid.org/0000-0003-0276-4517, vlad@icc.spbstu.ru*
[a]*Peter the Great St.Petersburg Polytechnic University, 19, Politechnicheskaya St., 195251, Saint-Petersburg, Russian Federation*
[b]*JetBrains Co. Ltd., 70, Building 1, Primorskiy Av., 197374, Saint-Petersburg, Russian Federation*

Introduction: *Over the last years program analysis methods were widely used for software quality assurance. Different types of program analysis require various levels of program representation, analysis methods, etc. Platforms that provide utilities to implement different types of analysis on their basis become very important because they allow one to simplify the process of development.* ***Purpose:*** *Development of a platform for analysis of JVM programs.* ***Results:*** *In this paper we present Kex, a platform for building program analysis tools for JVM bytecode. Kex provides three abstraction levels. First is Kfg, which is an SSA-based control flow graph representation for bytecode-level analysis and transformation. Second is a symbolic program representation called Predicate State, which consists of first order logic predicates that represent instructions of the original program, constraints, etc. The final level is SMT integration layer for constraint solving. It currently provides an interface for interacting with three SMT solvers.* ***Practical relevance:*** *We have evaluated our platform by considering two prototypes. First prototype is an automatic test generation tool that has participated in SBST 2021 tool competition. Second prototype is a tool for detection of automatic library integration errors. Both prototypes have proved that Kex can be used to implement competitive and powerful program analysis tools.*

Keywords *— program analysis, analysis platform, test generation, symbolic execution.*

For citation: Abdullin A. M., Itsykson V. M. Kex: A platform for analysis of JVM programs. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 30–43. doi:10.31799/1684-8853-2022-1-30-43

Introduction

Modern world is highly dependent on software: it controls almost every part of human life. Thus, errors in the modern-day software may lead to fatal consequences. To address that problem IT-industry adopts software quality assurance techniques.

Software quality assurance techniques could be divided into two main groups: manual techniques and automatic ones. Manual quality assurance techniques include software testing, code review, audits, etc. Those techniques have proven their effectiveness over time and are currently used in everyday development processes. However, they all share one significant weakness: manual quality assurance is very hard and time-consuming work [1].

Automatic software quality assurance techniques are trying to overcome that weakness. Alike manual techniques, automatic methods vary on the level of complexity and depth of the analysis: from simple and fast code smell detection [2] to resource intensive verification [3]. Over the last years, automatic quality assurance methods were widely used for automatic testing, automatic test generation, bug detection, etc. Most of the automatic quality assurance techniques are based on methods of static (e. g. symbolic execution [4, 5], bounded model checking [6], etc.) and dynamic (e. g. fuzzing [7], dynamic symbolic execution [8], concolic testing [9], etc.) program analysis. Many IT-companies are currently using program analysis methods as part of their everyday development process [10, 11].

For most widely used programming languages like Java, JavaScript, C/C++, etc. there already exists a large variety of tools for both static and dynamic analysis [12–15].

In this paper we present Kex (https://github.com/vorpal-research/kex), a platform for building various kinds of program analysis tools for Java Virtual Machine (JVM) based languages. Kex consists of three main components: Kfg library for JVM bytecode analysis and transformations, intermediate representation called Predicate State (PS) for symbolic program representation and constraint solving module based on satisfiability modulo theories (SMT) solvers. These modules allow one to build different types of analyses on top of Kex, both dynamic (based on bytecode instrumentation and execution) and static (based on symbolic execution and constraint solving). To showcase capabilities of Kex we have considered two prototypes of analysis tools: one for automatic test generation for Java language and the other for automatic integration errors detection. Evaluation results show that Kex can successfully be used to analyze JVM based languages on a variety of levels of depth, complexity and precision.

Related work

As we have mentioned earlier, there already exists a number of tools and frameworks for analysis of JVM bytecode. These frameworks differentiate by analyses they support which the underlying model of program representation inherently limits. Let us consider some of the most significant examples.

ASM [16] is an all-purpose Java bytecode analysis and manipulation framework that was introduced in 2002. The project is still under active development and the latest release of version 9.2 was in the summer of 2021. ASM provides a set of bytecode analyses and transformations and can be used both to modify existing classes and to dynamically generate new classes. ASM is focused on working with low-level representation of compiled classes and therefore is mainly used by many projects (https://asm.ow2.io/). The Kfg library also uses ASM for working with JVM bytecode.

Soot [17] is a Java bytecode optimization framework that was first introduced in 1999. Soot provides four intermediate representations for bytecode: *baf* — simplified stack based bytecode, *jimple* — 3-address code representation of bytecode, *shimple* — static single assignment (SSA) variation [18] of *jimple* and *grimp* — an aggregated version of *jimple* suitable for decompilation and code inspection. Each representation is more suited for its own kind of analyses and optimizations including points-to analysis, call-graph construction, data-flow analysis, etc. Both ASM and Soot are libraries which are mainly focused on bytecode-level optimizations and do not provide tools for more in-depth analysis.

Spoon [19] is a library for Java source code analysis and transformation. Spoon meta model consists of three parts:

— structural part contains the declarations of program elements (classes, interfaces, methods, etc.);

— code part contains executable Java code in the form of AST;

— reference part models references to program elements.

Limitations of Spoon come from its meta model. First, as it works at the source code level, it is only limited to work with one language, whereas bytecode level frameworks can work with any JVM based languages. Second, Spoon provides only one program model — AST — which is not always well-suited for various types of analyses.

JBSE [5] is a symbolic JVM for automated program analysis, verification and test generation. JBSE uses javassist [20] library to interact with the target classes, provides its own API for working with source code at a bytecode level and provides an implementation of *symbolic state* that represents the state of execution of a program. Symbolic state can be transformed into an SMT formula in smtlib2 format [21], which is then sent to an SMT solver to reason about reachability of that state. JBSE also provides utilities for automatically generating test cases that reach a given state using reflection [22]. Authors have also developed two tools on top of JBSE. SUSHI [23] is an automatic test case generator for Java programs that uses JBSE for symbolic execution and EvoSuite [24] for test generation, which allows it to generate tests that do not use reflection. TARDIS [25] is an extension of SUSHI that uses JBSE to perform concolic testing. Those tools confirm the applicability of JBSE; however, it still has some limitations. First, JBSE does not provide any utilities to work with more structured program representation rather than stack-based bytecode; hence, the symbolic state is also very close to low-level bytecode representation. Second, the internal structure of JBSE is more suited for easy usage of symbolic execution results, but it is hard to extend it.

JDQL [26] is a framework for Java static analysis that uses Datalog [27] query language for automatically detecting bad patterns in the program source code. JDQL works both with Java source code and JVM bytecode, provides utilities to perform flow analysis and intra-procedural analysis, and it is easy to extend with new error detectors. However, JDQL is limited in a sense that it is only suitable for a lightweight pattern recognition based static analysis and does not allow performing more precise and complex types of analyses.

As one can see, there already exist many frameworks for analysis of JVM programs (both on bytecode and source code level). Existing frameworks are well suited for building analysis tools at one specific program representation level. For example, symbolic execution engines do not provide access to underlying source code and bytecode analyses and manipulation frameworks do not provide utilities for more in-depth analysis. This limitation of the existing frameworks has inspired us to develop a new platform that will:

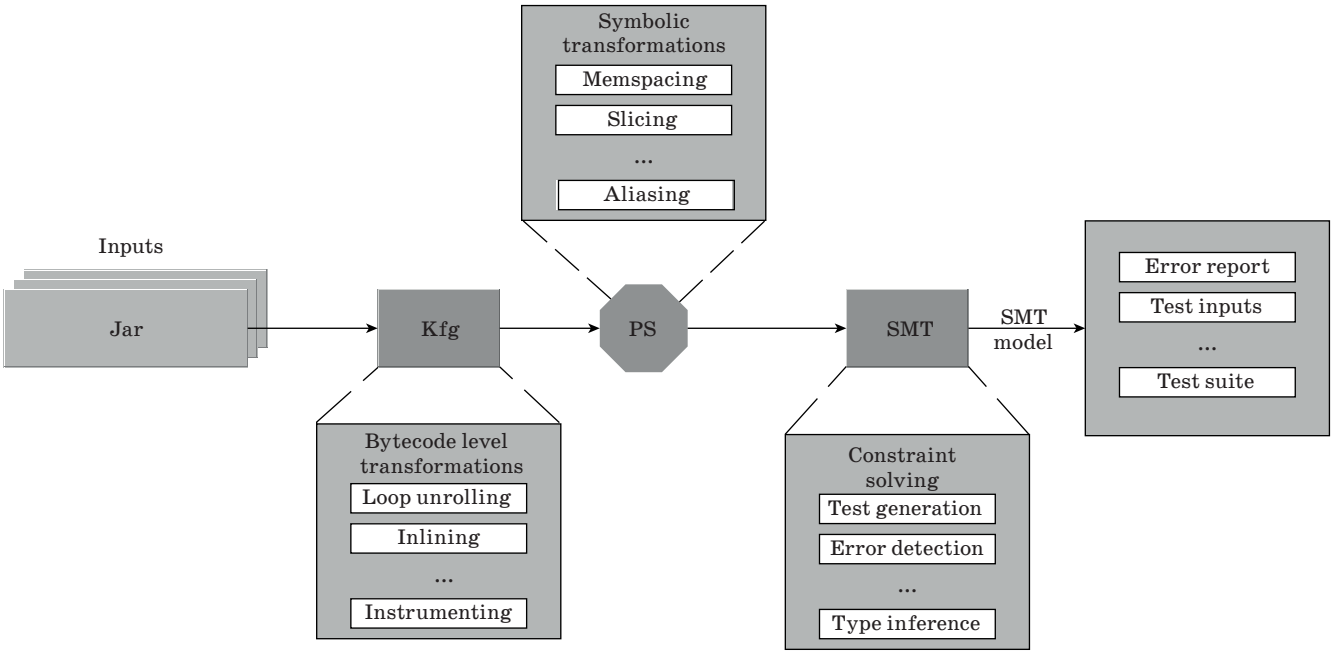
— provide utilities for various types of analyses (both static and dynamic);

— allow multi-level analysis;

— provide application programming interface (API) for constraint solving.

Kex in detail

Kex is a platform for analysis of JVM based languages. It takes a set of compiled classes and provides utilities to perform transformation and analyses on multiple levels of program representation: control flow graphs, PS and SMT formulae. Kex as-



■ ***Fig. 1.*** A high level overview of Kex

sumes a closed-world model during analysis, i. e., it has full access to all possible types, functions, etc. A high-level overview of Kex architecture can be found in Fig. 1. In this section we give a detailed description of every module of Kex.

Kfg: control flow graph for JVM bytecode

Program analysis requires having an informative and easy-to-use program model. JVM combines in itself features of stack machine and register machine: each execution frame has an operand stack and an array of local variables. The operand stack is used to provide operands for bytecode instructions and to receive results of their computations. The local variables array serves the same purpose as processor registers: to store quickly accessible data and to pass arguments for methods. While that model of computation is very effective for JVM purposes, it is not fitted for purposes of program analysis.

Kfg (https://github.com/vorpal-research/kfg) is a library for JVM bytecode analysis and transformation. Kfg builds control flow graphs (CFG) in SSA form [18] for each method of the target program. Kfg also provides utilities to create and modify classes and fields of a project. Kfg is built on top of the latest ASM version 9.2 — an all-purpose Java bytecode manipulation and analysis framework — and provides an API to directly access ASM representation for more low-level features. Currently Kfg supports JVM bytecode version 62 and lower (which corresponds to JVM version 18). Let us now consider the internal structure of Kfg in more detail.

Class management

The key concept of the Kfg is the *ClassManager*, which stores all the information about available classes and allows one to access those classes. Classes are the essential part of JVM and, therefore, Kfg: every project consists of a set of classes. Each class contains the following list of information: modifiers, superclass, interfaces, methods, etc. Each class also stores an instance of *ClassNode* — an ASM representation of class — allowing it to make low-level transformations.

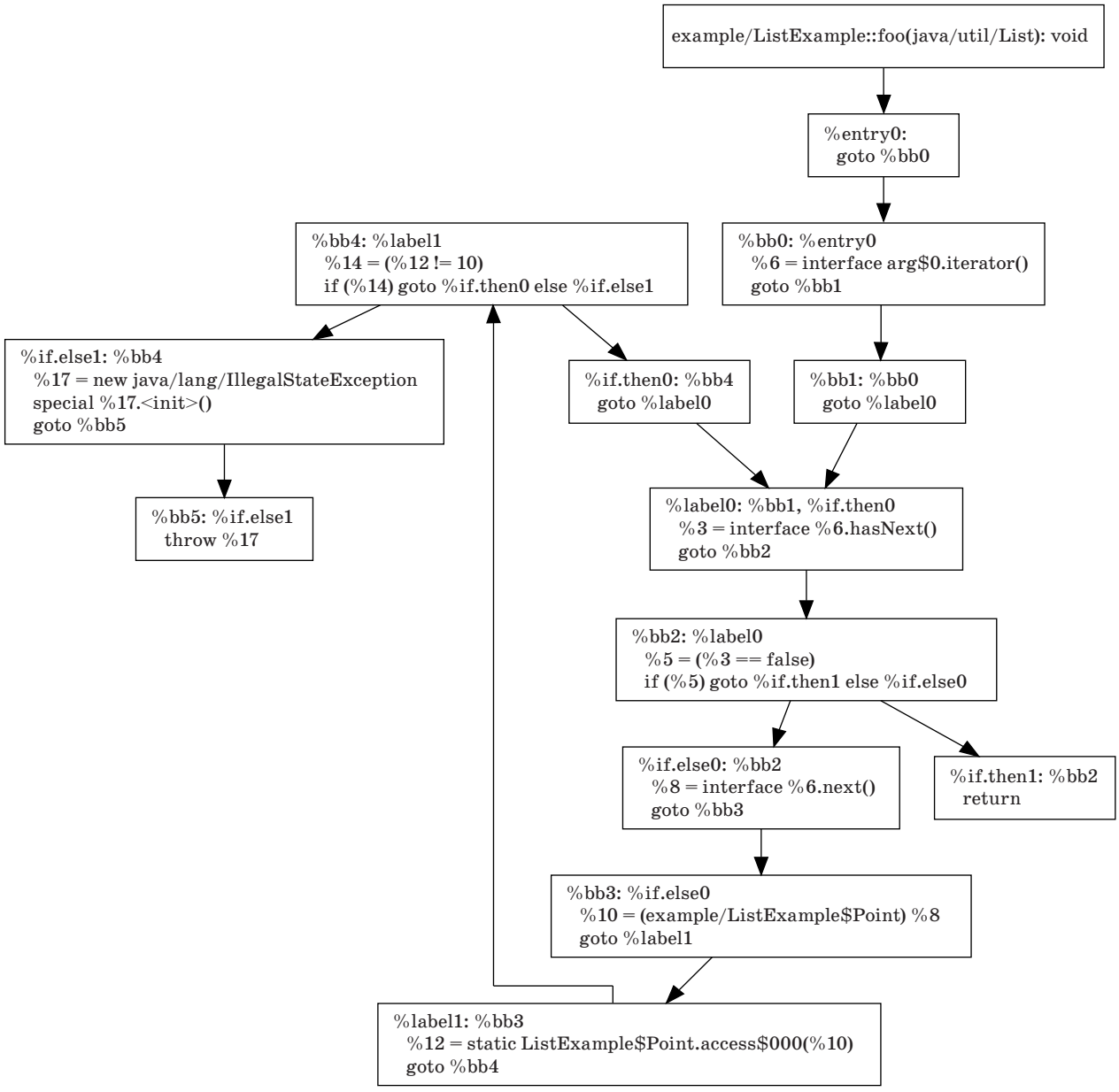
Kfg preserves connection between each class and its actual bytecode stored in the file system, thus allowing creating, modifying and updating the bytecode both on singular class level and on project level (e. g. modify jar files or directories with compiled sources). That connection is implemented through *Containers*.

To build a precise model of a project one needs to have access to all the libraries that it depends on. However, building that model for a large-scale project with many dependencies can be very resource intensive and even redundant in some cases. To have an ability to analyze projects without access to full class path, Kfg introduces an idea of *OuterClass*: a class which bytecode Kfg cannot access. When working with instances of *ConcreteClass* (i.e. class whose bytecode that Kfg has access to), Kfg checks validity and correctness of all operations. Downside of the *OuterClass* idea is that Kfg cannot guarantee correctness of the resulting bytecode and relies on the user to ensure it.

Type system of Kfg directly corresponds to JVM type system described in the JVM specification [28] and consists of integrals (boolean, byte, char, short, int and long), floating points (float and double), references (classes, arrays and null) and void type. Let us now consider the details of the control flow graph.

Methods and CFG

For each method of each *ConcreteClass* available to *ClassManager* Kfg builds a CFG in SSA form. An example of CFG built by Kfg is shown in Fig. 2. A control flow graph consists of basic blocks — a sequence of instructions that are executed one after other without any branching. Each basic block ends with terminating instruction; it can be simple jump, branching, return or throw. As one can see, basic blocks *%entry0, %bb0, %bb1* could have been united to a single basic block as they do not have any branching and seemingly are always directly following each other. However, CFG for JVM bytecode is more complex, as the instructions (and, therefore, basic blocks) of JVM bytecode may have *hidden* connections to exception catch blocks. JVM specification defines all the instructions that can potentially throw exceptions. Each basic block of Kfg ends either with branching instruction or with an exception throwing instruction. Figure 2 shows the examples of both of this cases: blocks *%bb2* and *%bb4* end with branch instruction (if — else); blocks



■ ***Fig. 2.*** An example of CFG built by Kfg

%bb0 and *%label0*, for example, end with simple jump instruction but the last "meaningful" instruction they contain is the call instruction, which potentially can throw an exception. To express those hidden connections, basic blocks also store a set of *handler* blocks in addition to a set of predecessors, successors and instructions. Basic blocks themselves are also divided into two categories:

— *BodyBlock* — default block that forms the CFG;

— *CatchBlock* — special block that handles thrown exceptions; that block does not have usual predecessors but a set of *thrower* blocks that it catches from; also, *CatchBlock* stores information about the catched exception type.

Values and instructions

Instructions of the Kfg operate on *Values*: representation of program variables. Values are divided into four categories: *this* reference, arguments, constants and instructions. Instruction set of Kfg is corresponding one-to-one to the instruction set of JVM bytecode with two exceptions:

— JSR instruction [28] is inlined before CFG building and therefore is not present in Kfg instructions set;

— Kfg adds a new *phi* instruction — a special instruction that represents φfunction of SSA form.

CFG analysis and modification

Along with the API to build a CFG model of a project and translate it back to JVM bytecode, Kfg provides an API to perform various analyses and transformations with the built model.

Kfg uses a visitor [29] pattern and provides a *NodeVisitor*, a *ClassVisitor* and a *MethodVisitor* allowing one to traverse all the classes, fields, methods and instructions of a project. In addition, Kfg also provides loop analysis for all of the methods of a project: information about each loop of a method is stored in a graph form. To simplify the work with loops Kfg also provides a *LoopVisitor*: an extension of *MethodVisitor* that allows to traverse all the loops of a method. *Pipelines* allow combining a set of visitors into a single instance that will apply each visitor to all the classes of *ClassManager* one after another.

At the instruction and value level Kfg implements the "user" pattern: each instruction, value and basic block contains a set of objects that it is used by. Any class that uses CFG elements should implement *BlockUser* or *ValueUser* interfaces. That idea was inspired by LLVM [30].

Kex currently uses Kfg for:

— loop canonicalization [31];

— loop unrolling;

— bytecode instrumentation on various levels;

— bytecode modification (e. g. all the *System.exit()* calls are replaced with a special *SystemExitCallException* to prevent JVM stopping during dynamic analysis);

— CFG modification, etc.

Predicate State representation

Predicate State is Kex's intermediate representation that is used to perform various types of analysis and that is designed to be easily converted into an SMT formula. This section describes details of PS implementation.

Basic PS structure

Predicate State is designed as a directed acyclic graph because SMT formulae cannot express loops. PS was originally introduced in Borealis bounded model checker [32]. Kex has adapted PS from Borealis and extended it to support Kfg instructions.

Predicate State is built from CFG and, therefore, CFG should be preprocessed in order to be convertible to PS. Preprocessing consists of two main steps:

— loop canonicalization;

— loop unrolling.

These two steps allow presenting a CFG in a form that is directly convertible to PS. Both of these operations are implemented as Kfg loop visitors. The format of PS in Backus — Naur form [33] is shown in listing 1 and an example of PS is shown in listing 2.

As one can see from listing 1, PS has three types:

— *BasicState* — PS represents a single basic block, basically just a list of predicates;

— *ChoiceState* — PS that represents branching, contains a list of branches (as PS);

— *ChainState* — PS that combines two states into a sequence, used to create full program representation from *BasicState* and *ChoiceState*.

Listing 1. PS format.

```
<PredicateState> ::= ChainState head:<PredicateState>
tail:<PredicateState>
    | ChoiceState choices:<ListOfPredicateStates>
    | BasicState data:<ListOfPredicates>

<ListOfPredicateStates> ::= <PredicateState>
<ListOfPredicateStates> | <empty>

<Predicate> ::= ArrayInitializerPredicate arrayRef:<Term>
value:<Term>
    | ArrayStorePredicate arrayRef:<Term> value:<Term>
    | CallPredicate lhv:<Term> call:<Term>
    | CatchPredicate throwable:<Term>
    | DefaultSwitchPredicate cond:<Term> cases:<ListOfTerms>
    | EnterMonitorPredicate monitor:<Term>
    | EqualityPredicate lhv:<Term> rhv:<Term>
    | ExitMonitorPredicate monitor:<Term>
    | FieldInitializerPredicate field:<Term> value:<Term>
    | FieldStorePredicate field:<Term> value:<Term>
    | GenerateArrayPredicate lhv:<Term> length:<Term>
generator:<Term>
    | InequalityPredicate lhv:<Term> rhv:<Term>
```

```
    | NewArrayPredicate lhv:<Term> dimensions:<ListOfTerms>
elementType:<Type>
    | NewPredicate lhv:<Term> type:<Type>

<Term> ::= ArgumentTerm index:Int type:<Type>
    | ArrayContainsTerm array:<Term> value:<Term>
    | ArrayIndexTerm array:<Term> index:<Term>
    | ArrayLengthTerm array:<Term>
    | ArrayLoadTerm arrayRef:<Term>
    | BinaryTerm op:<BinaryOp> lhv:<Term> rhv:<Term>
    | CallTerm owner:<Term> method:<Method>
arguments:<ListOfTerms>
    | CastTerm term:<Term>
    | CmpTerm op:<CmpOp> lhv:<Term> rhv:<Term>
    | ConstTerm
    | EqualsTerm lhv:<Term> rhv:<Term>
    | ExistsTerm start:<Term> end:<Term> body:<Term>
    | FieldTerm owner:<Term> fieldName:String
    | FieldLoadTerm field:<Term>
    | ForAllTerm start:<Term> end:<Term> body:<Term>
    | InstanceOfTerm term:<Term> type:<Type>
    | IteTerm cond:<Term> trueValue:<Term> falseValue:<Type>
    | LambdaTerm arguments:<ListOfTerms> body:<Term>
    | NegTerm term:<Term>
    | ReturnValueTerm method:<Method>
    | StaticClassRefTerm klass:<Type>
    | ValueTerm type:<Type> name:String

<ListOfTerms> ::= <Term> <ListOfTerms> | <empty>
```

Listing 2. PS example.

```
(
 @S kotlin/jvm/internal/Intrinsics.checkNotNullParameter(arg$0, 'a')
 @S %1 = arg$0.size()
 @S %3 = %1 != 2
 @P %3 = false
 @S %5 = arg$0.get(0)
 @S %7 = (%5 as example/ListExample$Point)
 @S %9 = %7.getX()
 @S %11 = %9 != 10
 @S %11 = false
 @S %13 = arg$0.get(1)
 @S %15 = (%13 as example/ListExample$Point)
 @S %17 = %15.getY()
 @S %19 = %17 != 11
 @S %19 = false
 @S %24 = new java/lang/IllegalStateException
 @S %23 = 'a'.toString()
 @S %24.<init>(%23)
 @S %26 = (%24 as java/lang/Thowable)
)
```

One may notice that current implementation of PS is limited because it does not handle try/catch blocks, i. e. exception handling is not supported. Potentially it can be implemented by adding *ChoiceState* at each predicate that leads to two branches: one to the next predicate in the program and one to a catch block that handles the exception. However, that will lead to an exponential growth of the state size. We consider adding exception control flow handling into PS as a part of our future work.

The design of PS is closer to SMT formulae than CFG and it introduces some of the concepts that are later passed on to an SMT solver. First, the PS model introduces a *memory* concept and explicitly separates expressions that change the memory from ones that do not: predicates and terms correspondingly. Thus, predicates are used to express actions that change the state and the memory of a program, e. g. *FieldStorePredicate* that writes value to some field. However, there are also predicates that allow us to express some additional constraints for a program. The type of predicate determines those properties:

— state — usual predicate that changes the state (and, therefore, the memory) of program;

— path — predicate that expresses the current path condition;

— assume — predicate that carries some additional information that Kex can assume is true;

— axiom — predicate that encodes some axioms that are always true (e. g. a class reference always being not null);

— require — predicate that encodes some properties that Kex should check for correctness.

The PS definition shows that most of the predicates directly correspond to Kfg instructions. However, there are some exceptions. For example, *FieldInitializerPredicate* that allows initializing the value of a field before actual program execution.

Terms mainly represent Kfg values and operations that do not change the memory state, e. g. arguments, constants, array index reads, etc. In JVM bytecode there are no ways to reference the memory address that holds the value of a field or an element of an array, one can only read the value stored in that location. However, during analysis one needs to differentiate between memory location and the value that it stores. For that reason, Kex adds two special pointer terms: *ArrayIndexTerm* and *FieldTerm*. To receive the value stored in a given location one needs to explicitly specify memory load action with *ArrayLoadTerm* and *FieldLoadTerm* correspondingly.

Type system of Kex extends the type system of Kfg by supporting special typing *ArrayIndexTerms* and *FieldTerms*. The type system consists of:

— integrals: bool, byte, char, short, int, long;

— reals: float and double;

— pointers — an equivalent of Kfg references:

  ○ object pointers;

  ○ array pointers;

  ○ references — types of array indexes and fields;

  ○ null;

— void.

PS modification and analysis

Analysis of a program suggests that one has an ability to traverse and modify the model, i. e. Predicate State. Kex provides a *Transformer* interface to traverse PS and a *RecollectingTransformer* interface to modify it. *Transformer* implements

CRTP pattern [34] and provides an API to dismantle each component of the PS and build it up again with the same or new structure. PS and its components are immutable and therefore if any transformer changes the state it returns a new copy of it.

Kex provides a set of transformer implementations:

— Stensgaard alias analysis [35];

— static backward slicing [36];

— constant propagation;

— inlining of various types: static fields initialization inlining, static method inlining, virtual method inlining (requires type resolving);

— reflection info inlining (e. g. Kotlin reflection provides a lot of useful type and nullity information);

— external information provider: e. g. annotation info inliner that adds method invocation info from JetBrains annotations (https://github.com/JetBrains/java-annotations), etc.

Symbolic execution using SMT solver

Kex uses SMT solvers for constraint solving and currently supports three solvers: Z3 [37], Boolector [38] and STP [39]. To simplify work with multiple solvers Kex uses automatically generated unified wrapper classes. Example of SMT wrapper API can be found in Fig. 3.

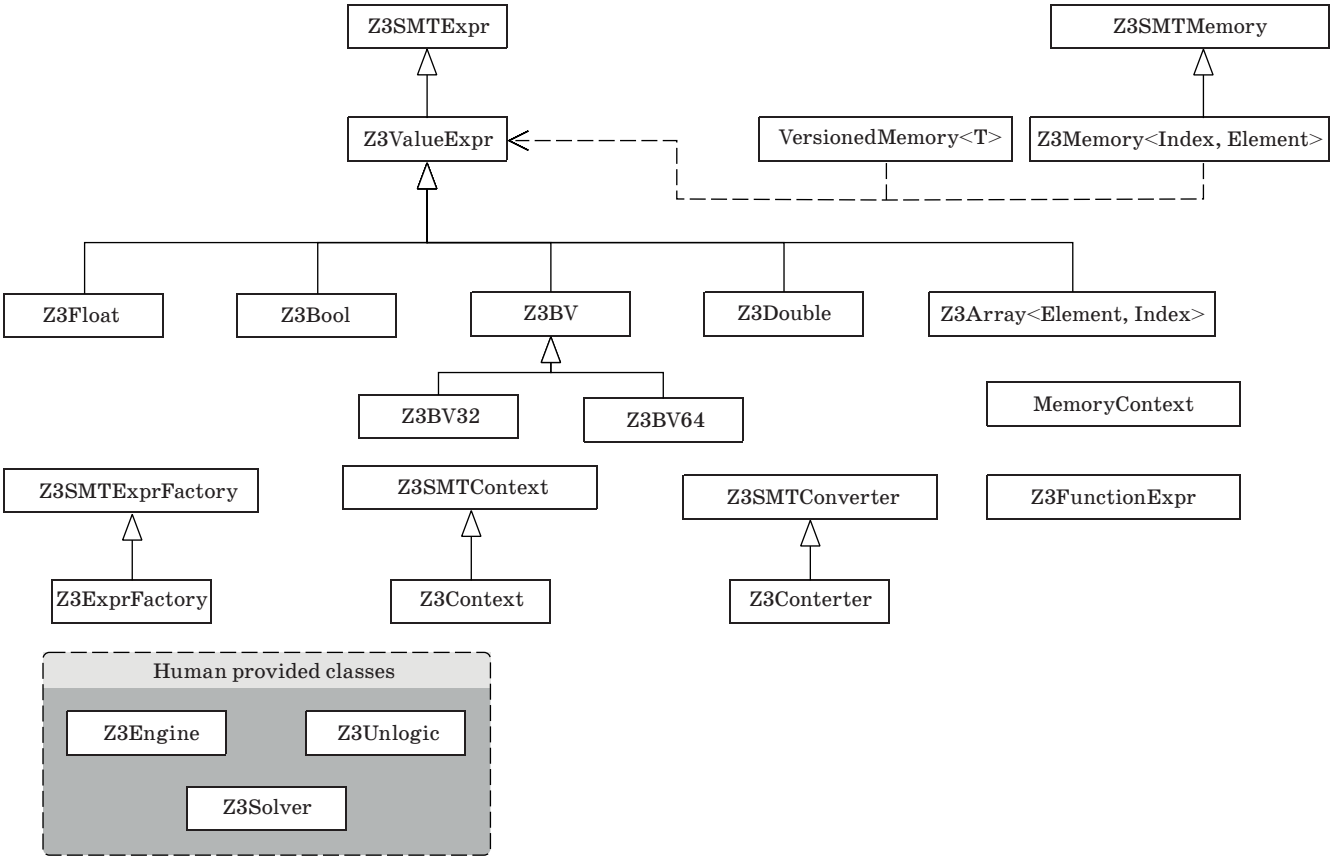
To add the new solver to Kex one needs to provide implementations of three classes: *Engine*, *Solver* and *Unlogic*. Engine class should provide bindings to the API of the solver. Solver class should implement methods that perform a query and return a model. Unlogic class should provide an interface to convert received model back into terms and predicates of PS.

In this section we will describe the model that Kex uses to express queries over PS in the SMT formula. First, however, let us describe the steps Kex uses to prepare PS.

Preparing PS

Predicate State preparation consists of two steps: reifying PS with additional information and complementing PS with necessary type information for SMT solving. The first step is optional and only used to give solver more information on the constraint solving: it inlines resolvable methods, includes available reflection and annotation information, propagates constants, etc. The result of the first step is the PS and the query over that state.

The main goal of the second step is to simplify the PS and the query so that SMT solver will be able



■ ***Fig. 3.*** Z3 solver wrapper classes

to solve it faster. It uses two techniques to reach its goal: memory spacing [40] and slicing.

Memory spacing is a technique that allows splitting all memory used in a program into a disjoint set of sub-memories. Each sub-memory is independent from the others and can be modeled separately. Each sub-memory is assigned a unique index, pointers referencing this memory are identified by the mentioned index. This reduces the complexity of solving the resulting formulae, as the disjoint set of memories decreases the search space SMT solver needs to work with.

Slicing is used to reduce the overall size of the PS. The main idea is to remove terms and predicates that are not "interesting" w.r.t. query from PS. The term is considered "interesting" if it affects or aliases any of the interesting terms. Aliasing is currently determined by Stensgaard alias analysis. Initial set of "interesting terms" contains all the variable (i. e. non-constant) terms from the query.

These preparation steps allow us to reinforce PS with additional information and simplify it w.r.t. SMT solving. Let us now consider how PS are encoded into an SMT formula.

Modeling program in SMT formulae

To be able to use SMT solver for constraint solving one needs to define a memory model suitable for representing the program and its variables as SMT formulae. In Kex we have used a memory model inspired by the work on bounded model checker Borealis. We have adapted its memory model to JVM and PS.

As was mentioned earlier, PS (like JVM bytecode) has several primitive data types: booleans, integers, floating point numbers. Each variable of a given type can be represented as an expression of corresponding SMT theory: booleans for boolean, bitvectors [41] for integers, floating point numbers [42] for float and double.

The more complex part, however, is modeling non-primitive data types: objects and arrays. To represent references in the heap we use a "property-based" memory model: memory is encoded as a collection of SMT arrays [43], each array corresponding to a disjoint partition of heap objects definitely not aliasing objects from other partitions. SMT arrays are immutable and each store operation returns a new version of the array. Therefore the memory model allows one to work with *versioned memory* i. e. one can potentially get the whole memory state of a program after execution of each instruction. Initial memory of the program is empty, it can be filled with special *FieldInitializer* and *ArrayInitializer* predicates.

This allows it to encode object references as 32-bit bitvector indices into their partition; arrays are represented as continuous chunks, with array reference pointing to its start index.

Object fields are represented in a similar fashion, using "property memories": each field is mapped to a separate SMT array, indexed by object references; to access field *x.y* one needs to work with property memory *typeOf(x).y* by index *x*.

Property memories also have one additional use case: they are used to calculate runtime type information of pointers. Resolving runtime type information is very important because it not only may affect control flow of a program (e. g. through *instanceof* instructions) but also is used to resolve virtual method calls. Each pointer variable of the program is assigned a special "type" property: each reference may be used as an index to this property memory to get its type. Kex analyses the program as a closed world model, therefore it can assign a constant to each defined type and encode subtyping via SMT axioms over a special *isSubtype* uninterpreted function:

$$\forall a,b \in types \begin{cases} isSubtype(a,b) = true, \\ if\ a \text{ is a subtype of } b \\ isSubtype(a,b) = false, \text{ otherwise} \end{cases}$$

All type-related operations in the program are expressed through *isSubtype*: casts and *instanceof* checks impose new constraints on the "type" property of the corresponding variable. That, together with the subtyping axioms, gives SMT solver enough information to correctly analyze types.

We have given a description of the memory model that Kex uses for symbolic execution. Given that, PS is a directed acyclic graph; its translation into SMT formula is straightforward, as predicates can be directly mapped to corresponding SMT expressions. One may vary the precision and complexity of SMT formulae by changing the depth of inlining and loop unrolling. An example of PS, query and corresponding SMT formula can be found in Fig. 4.

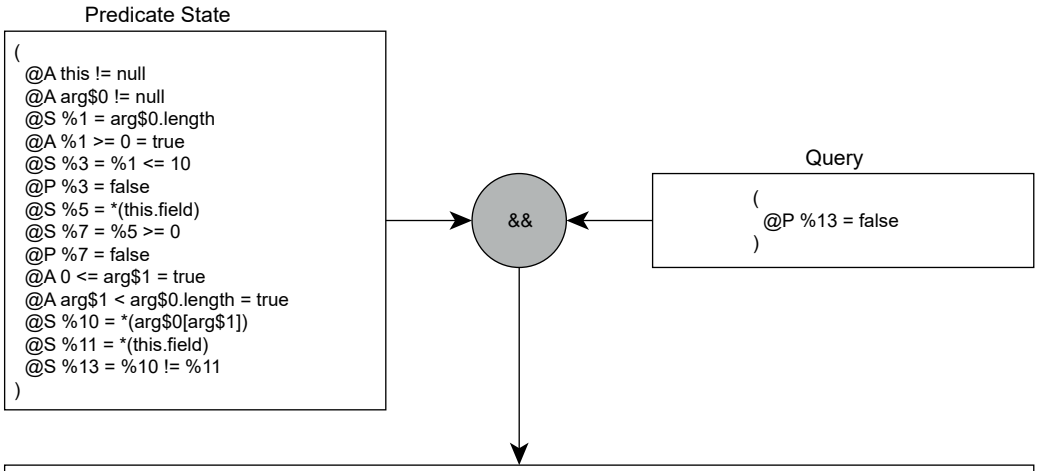
After encoding PS and query as SMT formulae Kex performs a request to SMT solver. SMT solver can return three types of answers:

— SAT — formula is satisfiable, solver also returns an SMT model containing counterexample that makes formula satisfiable;

— UNSAT — formula is unsatisfiable, solver also may return an unsatisfiable core [44], i. e. a minimal set of clauses that makes the formulae unsatisfiable;

— UNKNOWN — unknown result, returned if solver is terminated by timeout.

Depending on the query type, these results can be interpreted differently. However, if the formula is satisfiable, one needs to be able to raise the program state encoded in the SMT model to a higher level. Let us now describe how that is performed.




```
(declare-fun isSubtype ((_ BitVec 32) (_ BitVec 32)) Bool)
(declare-fun __word__property__type0!0 () (Array (_ BitVec 32) (_ BitVec 32)))
(declare-fun arg$0 () (_ BitVec 32))
(declare-fun __word__property__length0!3 () (Array (_ BitVec 32) (_ BitVec 32)))
(declare-fun |int[]!2| () (_ BitVec 32))
(declare-fun this () (_ BitVec 32))
(declare-fun PSExample!1 () (_ BitVec 32))
(declare-fun %13 () Bool)
(declare-fun %11 () (_ BitVec 32))
(declare-fun %10 () (_ BitVec 32))
(declare-fun __word__property__PSExample.field0!4 () (Array (_ BitVec 32) (_ BitVec 32)))
(declare-fun arg$1 () (_ BitVec 32))
(declare-fun __array__0!5 () (Array (_ BitVec 32) (Array (_ BitVec 32) (_ BitVec 64))))
(declare-fun %7 () Bool)
(declare-fun %5 () (_ BitVec 32))
(declare-fun %3 () Bool)
(declare-fun %1 () (_ BitVec 32))
(assert
  (and (not (= this #x00000000))
       (not (= arg$0 #x00000000))
       (= %1 (select __word__property__length0!3 arg$0))
       (bvsle #x00000000 %1)
       (= %3 (bvsle %1 #x0000000a))
       (not %3)
       (= %5 (select __word__property__PSExample.field0!4 this))
       (= %7 (bvsle #x00000000 %5))
       (not %7)
       (bvsle #x00000000 arg$1)
       (not (bvsle (select __word__property__length0!3 arg$0) arg$1))
       (= %10 ((_ extract 31 0) (select (select __array__0!5 arg$0) arg$1)))
       (= %11 (select __word__property__PSExample.field0!4 this))
       (= %13 (not (= %10 %11)))
       (bvsle #x00000000 (select __word__property__length0!3 arg$0))
       (not (bvsle #x000003e8 (select __word__property__length0!3 arg$0)))
       (bvsle #x00000000 this)
       (or (= this #x00000000)
           (= (select __word__property__type0!0 this)
              PSExample!1))
       (bvsle #x00000000 (select __word__property__type0!0 this))
       (not (bvsle #x3fffffff this))
       (bvsle #x00000000 arg$0)
       (or (= arg$0 #x00000000)
           (= (select __word__property__type0!0 arg$0) |int[]!2|))
       (bvsle #x00000000 (select __word__property__type0!0 arg$0))
       (bvsle #x00000000 (select __word__property__length0!3 arg$0))
       (not (bvsle #x3fffffff arg$0))
    )
  )
(assert (and true
     (= (isSubtype PSExample!1 PSExample!1) true)
     (= (isSubtype PSExample!1 |int[]!2|) false)
     (= (isSubtype |int[]!2| PSExample!1) false)
     (= (isSubtype |int[]!2| |int[]!2|) true))
  )
(assert (not %13))
(check-sat)
```

■ ***Fig. 4.*** An example of PS, query and SMT formula

Interpreting SMT model

An SMT model allows one to evaluate concrete values of formula expressions. In the context of program analysis those concrete values describe some interesting program state: counterexample that triggers a bug, test case that covers some branch, etc. Thus, to be useful, the SMT model eventually needs to be converted into Java objects or Java code that creates those objects. In Kex that conversion process consists of two steps: translating the model into terms and building Java objects from those terms.

The first step is straightforward: using special *unlogic* functions of each solver, Kex converts constants from the SMT model into constant terms. Kex model consists of three components:

— assignments — a map where each variable of a program is assigned a constant value evaluated from SMT model;

— memory shapes — each shape contains two memory states: initial and final. Each memory state maps concrete integer addresses into constants;

— type map — a map where each type of a program is assigned constant integer value. Type map is later used to evaluate runtime types of variables from type property memory.

An example of the Kex model corresponding to the SMT model is given in listing 3.

Listing 3. An example Kex model.

```
Model {
 this = 32768
 arg$0 = 1
 %1 = 35
 %3 = false
 %5 = -1
 this.field = 32768
 %7 = false
 arg$1 = 0
 %10 = -1
 %11 = -1
 %13 = false
 (1)<0> = 0
 (32768)<0> = 0
 (1073741823)<0> = 0
 PSExample.field(32768)<0> = -1
 length(1)<0> = 35
 type(1)<0> = 3
 type(32768)<0> = 4
 type(1073741823)<0> = 2
}
```

Although the Kex model is not very illustrative, it still captures the whole program state. For further transformation and analysis, Kex transforms SMT model into *descriptors*. Descriptors are used to represent the object shape; one may consider them trees that capture (nested) object states. The descriptor format for the JVM platform is given in listing 4.

Listing 4. JVM descriptor format.

```
<Descriptor> ::= “CostantDescriptor”
      | “ObjectDescriptor” fields:<ListOfFields>
      | “ArrayDescriptor” elements:<ListOfElements>
      | “StaticFieldDescriptor” field:<Field>

<ConstantDescriptor> ::= “NullDescriptor”
      | “BoolDescriptor” value:Boolean
      | “ByteDescriptor” value:Byte
      | “ShortDescriptor” value:Short
      | “CharDescriptor” value:Char
      | “IntDescriptor” value:Int
      | “LongDescriptor” value:Long
      | “FloatDescriptor” value:Float
      | “DoubleDescriptor” value:Double

<Field> ::= name:String klass:Class value:<Descriptor>

<Element> ::= index:Int value:<Descriptor>

<ListOfFields> ::= <Field> <ListOfFields> | <empty>

<ListOfElements> ::= <Element> <ListOfElements> | <empty>
```

Additionally Kex is able to build Java objects from the descriptors if needed. Kex collects all the variables from the program and builds Java objects for those projects using following algorithm:

— if a variable has primitive type, create a primitive Java variable with corresponding value;

— if a variable is an object, resolve its runtime type (using type map) and create Java object of resolved type (using Java reflection utilities);

— if a variable is an array, resolve its runtime type (using type map) and create Java array of resolved type (using Java reflection utilities);

— if a variable is a field, recursively create a Java object corresponding to its value and set the field value of an object (using Java reflection utilities);

— if a variable is an array element, recursively create a Java object corresponding to its value and set the element value of an array (using Java reflection utilities).

Kex also has techniques to generate a test case that recreates a program state corresponding to the SMT model, but its implementation details are left outside of this work.

Evaluation of Kex platform

The evaluation of our platform consists of two parts. First part is the qualitative comparison of Kex platform with other analogues considered earlier. Second part is the evaluation of Kex applicability for developing program analysis tools. In this part we will consider two prototypes of program analysis tools that were developed based on Kex platform.

Qualitative comparison with analogues

To evaluate our platform we decided to compare it with five other analogues, that were previously considered, based on seven criteria:

— source artifact: what artifacts does the tool takes as an input;

— source manipulation and transformation: does the tool provides utilities for transformation of the input sources;

— behavioral representation: whether tool provides behavioral program representation (like CFG, SSA, etc.) rather than simple stack-based bytecode;

— symbolic representation: does the tool provide a symbolic representation of a program that can be used for more in-depth analysis;

— constraint solving: does the tool provides API to work with any kind of constraint solvers;

— static analysis utilities: does the tool has built in utilities for static program analysis;

— dynamic program analysis: does the tool has built in utilities for static program analysis.

The results of the comparison can be found in the Table. As one can see from the results, ASM and Soot frameworks are libraries which are mainly focused on bytecode-level optimizations and do not provide tools for more in-depth analysis. Spoon is similar to ASM and Soot except that is concentrates on the Java source code analysis. JBSE is similar to Kex in almost every criteria; however, its main weakness is that it does not provide any utilities to work with behavioral program representations like CFG. JDQL is only tool that supports both source code and bytecode analysis and allows one to solve queries over program variables using Datalog. However, it is only suitable for a lightweight pattern recognition based static analysis and does not allow performing more precise and complex types of analyses. Judging by the results Kex is the tool that fits the most criteria; the only weakness is that Kex does not support source code analysis. That decision was intentional, because as source code analysis may provide more information about program (e. g. generics), it bounds the tool to only one programming language (or requires too much infrastructure for working with multiple languages).

Evaluation of prototypes

To evaluate our platform we have implemented a prototype of an automatic test generation tool for Java language based on Kex infrastructure. The prototype uses symbolic execution to analyze control flow graphs of the program under test (PUT) and produces interesting symbolic inputs for each basic block of PUT. Those symbolic inputs are then converted into JUnit test cases (either in Java or in Kotlin language). Prototype currently supports two modes of test case generation: basic, which generates reflection based test cases, and advanced, that tries to generate test cases using only public API's of the PUT. We have participated in the SBST 2021 Tool Competition [45, 46] with the described prototype. With an overall score of 44.21, Kex ranked fifth. Thorough analysis of the results has shown that the prototype had many technical issues due to a low degree of maturity of the project. On the *guava* project, Kex was able to reach ~20% line coverage, which is competitive

■ Qualitative comparison of Kex with analogues

Criteria	ASM	Soot	Spoon	JBSE	JDQL	Kex
Source artifact	JVM bytecode	JVM bytecode	Java	JVM bytecode	Java or JVM bytecode	JVM bytecode
Source manipulation and transformation	+	+	+	+	–	+
Behavioral representation	–	+	–	–	–	+
Symbolic representation	–	–	–	+	–	+
Constraint solving	–	–	–	SMTLib2 formulae for SMT solvers	Datalog queries	API for Z3, Boolector and STP SMT solvers
Static analysis utilities	–	–	–	+	+	+
Dynamic analysis utilities	–	–	–	+	–	+

with the results of other participating tools. After resolving all the technical issues with the prototype, it was able to reach ~25% average line coverage on the whole SBST 2021 competition benchmark (https://github.com/vorpal-research/kex/tree/sbst-21). We consider that as significant improvement.

Another application of Kex platform is Spider [47]. The authors had built a tool that allows them to find library integration errors using static analysis methods. Authors enrich the source code of external libraries with formal specifications written in LibSL specification language [48]. They use Kfg library to inject specification automata into the original library classes. All the necessary checks are marked with calls to a Kex intrinsics (https://github.com/vorpal-research/kex-intrinsics) library. Implemented analysis module finds the library API function calls and checks their conditions. If the condition can be false then intrinsic call is reachable, and the error occurs.

We conclude that Kex is an applicable and extendable platform for building various types of program analysis, both static and dynamic.

Conclusion

In this paper we presented a platform for analysis of Java programs called Kex. We have described all the main components of Kex, their implementation details and external APIs. Kex can be used to build tools for various types of program analysis, both lightweight and complex. During evaluation, we have considered two prototypes of program analysis tools: one for automatic test generation and one for integration errors detection. Evaluation has shown that Kex is applicable for creating program analysis tools for JVM platform.

In the future we plan to further work on improving capabilities of Kex. In terms of capabilities of Kex as a platform, we want to implement an exception handling mechanism in PS and improve lambda function support both on PS level and on SMT formulae level. Another area of our future work is the development of tools based on Kex. Currently we have two main priorities for the future: improve our automatic test generation tool for Java language and participate in the SBST 2022 competition and develop a concolic testing tool based on Kex.

References

1. Sharma R. M. Quantitative analysis of automation and manual testing. *International Journal of Engineering and Innovative Technology*, 2014, no. 1, pp. 252–257.
2. De Stefano M., Gambardella M. S., Pecorelli F., Palomba F., De Lucia A. cASpER: A Plug-in for automated code smell detection and refactoring. *Proceedings of the International Conference on Advanced Visual Interfaces*, 2020, pp. 1–3. doi:10.1145/3399715.3399955
3. Jhala R., Majumdar R. Software model checking. *ACM Computing Surveys (CSUR)*, 2009, no. 4, pp. 1–54. doi:10.1145/1592434.1592438
4. Zhang T., Wang P., Guo X. A survey of symbolic execution and its tool KLEE. *Procedia Computer Science*, 2020, pp. 330–334. doi:10.1016/j.procs.2020.02.090
5. Braione P., Denaro G., Pezzè M. JBSE: A symbolic executor for Java programs with complex heap inputs. *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2016, pp. 1018–1022. doi:10.1145/2950290.2983940
6. Gadelha M. R., Menezes R. S., Cordeiro L. C. ESBMC 6.1: automated test case generation using bounded model checking. *International Journal on Software Tools for Technology Transfer*, 2021, no. 6, pp. 857–861. doi:10.1007/s10009-020-00571-2
7. Klees G., Ruef A., Cooper B., Wei S., Hicks M. Evaluating fuzz testing. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 2123–2138. doi:10.1145/3243734.3243804
8. Zhang L., Xie T., Zhang L., Tillmann N., De Halleux J., Mei H. Test generation via dynamic symbolic execution for mutation testing. *2010 IEEE International Conference on Software Maintenance*, 2010, pp. 1–10. doi:10.1109/icsm.2010.5609672
9. Sen K. Concolic testing. *Proceedings of the twenty-second IEEE/ACM International Conference on Automated Software Engineering*, 2007, pp. 571–572. doi:10.1145/1321631.1321746
10. Ayewah N., Pugh W., Hovemeyer D., Morgenthaler J. D., Penix J. Using static analysis to find bugs. *IEEE Software*, 2008, no. 5, pp. 22–29. doi:10.1109/ms.2008.130
11. Calcagno C., Distefano D., Dubreil J., Gabi D., Hooimeijer P., Luca M., O'Hearn P., Papakonstantinou I., Purbrick J., Rodriguez D. Moving fast with software verification. *NASA Formal Methods Symposium*, Cham, 2015, pp. 3–11. doi:10.1007/978-3-319-17524-9_1
12. Nielsen B. B., Møller A. Value Partitioning: A lightweight approach to relational static analysis for JavaScript. *34th European Conference on Object-Oriented Programming (ECOOP 2020)*, 2020, pp. 16:1–16:28.
13. Böhme M., Pham V. T., Nguyen M. D., Roychoudhury A. Directed greybox fuzzing. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2329–2344. doi:10.1145/3133956.3134020
14. Kroening D., Tautschnig M. CBMC–C bounded model checker. *International Conference on Tools and Algo-*

rithms for the Construction and Analysis of Systems, Berlin, 2014, pp. 389–391. doi:10.1007/978-3-642-54862-8_26
15. Visser W., Păsăreanu C. S., Khurshid S. Test input generation with Java PathFinder. *Proceedings of the 2004 ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2004, pp. 97–107. doi:10.1145/1007512.1007526
16. Bruneton E., Lenglet R., Coupaye T. ASM: A code manipulation tool to implement adaptable systems. *Adaptable and Extensible Component Systems*, 2002, no. 19. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.5769 (accessed 5 December 2021).
17. Vallée-Rai R., Co P., Gagnon E., Hendren L., Lam P., Sundaresan V. Soot: A Java bytecode optimization framework. *CASCON First Decade High Impact Papers*, 2010, pp. 214–224. doi:10.1145/1925805.1925818
18. Cytron R., Ferrante J., Rosen B. K., Wegman M. N., Zadeck F. K. Efficiently computing static single assignment form and the control dependence graph. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 1991, no. 4, pp. 451–490. doi:10.1145/115372.115320
19. Pawlak R., Monperrus M., Petitprez N., Noguera C., Seinturier L. Spoon: A library for implementing analyses and transformations of java source code. *Software: Practice and Experience*, 2016, no. 9, pp. 1155–1179. doi:10.1002/spe.2346
20. Shigeru C. Load-time structural reflection in Java. *14th European Conference on Object-Oriented Programming (ECOOP 2000)*, 2000, pp. 313–336. doi:10.1007/3-540-45102-1_16
21. Barrett C., Stump A., Tinelli C. The SMT-LIB Standard: Version 2.0. *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories*, Edinburgh, 2010. 14 p.
22. Demers F. N., Malenfant J. Reflection in logic, functional and object-oriented programming: A short comparative study. *Proceedings of the IJCAI*, 1995, pp. 29–38.
23. Braione P., Denaro G., Mattavelli A., Pezzè M. SUSHI: a test generator for programs with complex structured inputs. *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, 2018, pp. 21–24. doi:10.1145/3183440.3183472
24. Fraser G., Arcuri A. Evosuite: automatic test suite generation for object-oriented software. *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, 2011, pp. 416–419. doi:10.1145/2025113.2025179
25. Braione P., Denaro G. SUSHI and TARDIS at the SBST2019 tool competition. *2019 IEEE/ACM 12th International Workshop on Search-Based Software Testing (SBST)*, 2019, pp. 25–28. doi:10.1109/sbst.2019.00016
26. Saxena A., Soundrapandian P. D., Sharma V. S., Kaulgud V. JDQL: A framework for Java Static Analysis. *Proceedings of the 9th India Software Engineering Conference*, 2016, pp. 136–140. doi:10.1145/2856636.2856645
27. Huang S. S., Green T. J., Loo B. T. Datalog and emerging applications: an interactive tutorial. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, 2011, pp. 1213–1216. doi:10.1145/1989323.1989456
28. Lindholm T., Yellin F., Bracha G., Buckley A. *The Java virtual machine specification*. Pearson Education, 2014. Available at: https://docs.oracle.com/javase/specs/jvms/se8/html/index.html (accessed 5 December 2021).
29. Palsberg J., Jay C. B. The essence of the Visitor pattern. *Proceedings. The Twenty-Second Annual International Computer Software and Applications Conference (Compsac'98)*, 1998, pp. 9–15. doi:10.1109/cmpsac.1998.716629
30. Lattner C., Adve V. LLVM: A compilation framework for lifelong program analysis & transformation. *International Symposium on Code Generation and Optimization*, 2004, pp. 75–86. doi:10.1109/cgo.2004.1281665
31. Kaushik M. D. *Loop Fusion in LLVM Compiler*. Bach. of eng. Diss., Visvesvaraya Technological University, 2015. 39 p.
32. Akhin M., Belyaev M., Itsykson V. Borealis bounded model checker: The coming of age story. *Present and Ulterior Software Engineering*, Cham, 2017, pp. 119–137. doi:10.1007/978-3-319-67425-4_8
33. McCracken D. D., Reilly E. D. Backus-naur form (bnf). *Encyclopedia of Computer Science*, 2003, pp. 129–131.
34. Coplien J. O. Curiously recurring template patterns. *C++ gems*, May 1996, pp. 135–144.
35. Steensgaard B. Points-to analysis in almost linear time. *Proceedings of the 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1996, pp. 32–41. doi:10.1145/237721.237727
36. Weiser M. Program slicing. *IEEE Transactions on Software Engineering*, 1984, no. 4, pp. 352–357. doi:10.1109/tse.1984.5010248
37. De Moura L., Bjørner N. Z3: An efficient SMT solver. *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Berlin, 2008, pp. 337–340. doi:10.1007/978-3-540-78800-3_24
38. Niemetz A., Preiner M., Biere A. Boolector 2.0. *Journal on Satisfiability, Boolean Modeling and Computation*, 2014, no. 1, pp. 53–58. doi:10.3233/sat190101
39. Ganesh V., Dill D. L., A decision procedure for bit-vectors and arrays. *Computer Aided Verification, 19th International Conference*, Berlin, 2007, pp. 519–531. doi:10.1007/978-3-540-73368-3_52
40. Kapus T., Cadar C. A segmented memory model for symbolic execution. *Proceedings of the 2019 27th*

ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2019, pp. 774–784. doi:10.1145/3338906.3338936

41. Jha S., Limaye R., Seshia S. A. Beaver: Engineering an efficient SMT solver for bit-vector arithmetic. *International Conference on Computer Aided Verification*, Berlin, 2009, pp. 668–674. doi:10.1007/978-3-642-02658-4_53
42. Rümmer P., Wahl T. An SMT-LIB theory of binary floating-point arithmetic. *International Workshop on Satisfiability Modulo Theories (SMT)*, 2010, p. 151.
43. Stump A., Barrett C. W., Dill D. L., Levitt J. A decision procedure for an extensional theory of arrays. *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science*, 2001, pp. 29–37. doi:10.1109/lics.2001.932480
44. Guthmann O., Strichman O., Trostanetski A. Minimal unsatisfiable core extraction for SMT. *2016 Formal Methods in Computer-Aided Design (FMCAD)*, 2016, pp. 57–64. doi:10.1109/fmcad.2016.7886661
45. Panichella S., Gambi A., Zampetti F., Riccio V. SBST tool competition 2021. *2021 IEEE/ACM 14th International Workshop on Search-Based Software Testing (SBST)*, 2021, pp. 20–27. doi:10.1109/sbst52555.2021.00011
46. Abdullin A., Akhin M., Belyaev M. Kex at the 2021 SBST tool competition. *2021 IEEE/ACM 14th International Workshop on Search-Based Software Testing (SBST)*, 2021, pp. 32–33. doi:10.1109/sbst52555.2021.00014
47. Feofilaktov V., Itsykson V. M. SPIDER: Specification-based integration defect revealer. *International Conference on Tools and Methods for Program Analysis*, Tomsk, 2021. Available at: https://arxiv.org/abs/2202.03943 (accessed 9 February 2021).
48. Itsykson V. Partial specifications of libraries: Applications in software engineering. *International Conference on Tools and Methods for Program Analysis*, Cham, 2019, pp. 3–25. doi:10.1007/978-3-030-71472-7_1

УДК 004.05
doi:10.31799/1684-8853-2022-1-30-43

Kex: платформа для анализа JVM-программ

А. М. Абдуллин[а,б], аспирант, ассистент, orcid.org/0000-0002-9669-2587
В. М. Ицыксон[а,б], канд. техн. наук, доцент, orcid.org/0000-0003-0276-4517, vlad@icc.spbstu.ru
[а]Санкт-Петербургский политехнический университет Петра Великого, Политехническая ул., 19, Санкт-Петербург, 195251, РФ
[б]JetBrains Co. Ltd., Приморский пр., 70, к. 1, Санкт-Петербург, 197374, РФ

Введение: методы статического и динамического анализа программ все чаще используются для проверки качества программного обеспечения. Однако разные виды анализа программ требуют работы с разными моделями представления программ, методами анализа и т. д. Возросла важность платформ для создания инструментов анализа программ, так как они позволяют упростить и ускорить процесс разработки. **Цель:** разработать платформу для анализа JVM-программ. **Результаты:** разработана платформа Kex для построения инструментов анализа программ, компилирующихся в JVM-байткод. Kex предоставляет три уровня абстракции. Первый уровень — библиотека Kfg — реализует граф потока управления в форме статического однократного присваивания для анализа и трансформации JVM-байткода. Второй уровень — символьное представление программы, называемое Predicate State, которое состоит из предикатов логики первого порядка, соответствующих инструкциям программы, контрактам, дополнительным ограничениям и т. д. Третий уровень — интерфейс для создания и работы с SMT-формулами, позволяющий решать задачи выполнимости. Интерфейс в данный момент поддерживает взаимодействие с тремя SMT-решателями. **Практическая значимость:** платформа Kex использовалась при разработке двух инструментов: автоматической генерации тестов для языка Java, который был подан на соревнования SBST 2021, и автоматического поиска ошибок интеграции библиотек. Оба этих прототипа показали, что платформа Kex может быть использована для разработки инструментов автоматического анализа программ.

Ключевые слова — анализ программ, платформа для анализа программ, автоматическая генерация тестов, символьное исполнение.

Для цитирования: Abdullin A. M., Itsykson V. M. Kex: A platform for analysis of JVM programs. *Информационно-управляющие системы*, 2022, № 1, с. 30–43. doi:10.31799/1684-8853-2022-1-30-43
For citation: Abdullin A. M., Itsykson V. M. Kex: A platform for analysis of JVM programs. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 30–43. doi:10.31799/1684-8853-2022-1-30-43

UDC 003.26
doi:10.31799/1684-8853-2022-1-44-53

Articles

A novel method for developing post-quantum digital signature algorithms on non-commutative associative algebras

A. A. Moldovyan[a], *Dr. Sc., Tech., Professor, Chief Researcher, orcid.org/0000-0001-5480-6016, maa1305@yandex.ru*
D. N. Moldovyan[a], *PhD, Tech., Research Fellow, orcid.org/0000-0001-5039-7198*
N. A. Moldovyan[a], *Dr. Sc., Tech., Professor, Chief Researcher, orcid.org/0000-0002-4483-5048*
[a]*St. Petersburg Federal Research Center of the RAS, 39, 14th Line, V. O., 199178, Saint-Petersburg, Russian Federation*

Introduction: *Development of practical post-quantum signature algorithms is a current challenge in the area of cryptography. Recently, several candidates on post-quantum signature schemes, in which the exponentiation operations in a hidden commutative group contained in a non-commutative algebra is used, were proposed. Search for new mechanisms of using a hidden group, while developing signature schemes resistant to quantum attacks, is of significant practical interest.* ***Purpose:*** *Development of a new method for designing post-quantum signature algorithms on finite non-commutative associative algebras.* ***Results:*** *A novel method for developing digital signature algorithms on non-commutative algebras. A new four-dimensional finite non-commutative associative algebra set over the ground field GF(p) have been proposed as algebraic support of the signature algorithms. To provide a higher performance of the algorithm, in the introduced algebra the vector multiplication is defined by a sparse basis vector multiplication table. Study of the algebra structure has shown that it can be represented as a set of commutative subalgebras of three different types, which intersect exactly in the set of scalar vectors. Using the proposed method and introduced algebra, a new post-quantum signature scheme has been designed. The introduced method is characterized in using one of the elements of the signature (e, S) in form of the four-dimensional vector* $\mathbf{S}$ *that is computed as a masked product of two exponentiated elements* $\mathbf{G}$ *and* $\mathbf{H}$ *of a hidden commutative group:* $\mathbf{S} = \boldsymbol{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{r}\mathbf{C}^{-1}$*, where non-permutable vectors* $\mathbf{B}$ *and* $\mathbf{C}$ *are masking multipliers; the natural numbers n and r are calculated depending on the signed document M and public key. The pair* $\langle \mathbf{G}, \mathbf{H} \rangle$ *composes a minimum generator systems of the hidden group. The signature verification equation has the form* $\mathbf{R} = (\mathbf{Y}_1\mathbf{S}\mathbf{Z}_1)^{e}(\mathbf{Y}_2\mathbf{S}\mathbf{Z}_2)^{e^2}$*, where pairwise non-permutable vectors* $\mathbf{Y}_1$, $\mathbf{Z}_1$, $\mathbf{Y}_2$, *and* $\mathbf{Z}_2$ *are element of the public key and natural number e that is computed depending on the value M and the vector* $\mathbf{R}$. ***Practical relevance:*** *Due to sufficiently small size of public key and signature and high performance, the developed digital signature scheme represents interest as a practical post-quantum signature algorithm. The introduced method is very attractive to develop a post-quantum digital signature standard.*

Keywords *— post-quantum cryptoschemes, computer security, digital signature, discrete logarithm problem, finite non-commutative algebras, associative algebras, cyclic groups, multidimensional cyclicity.*

For citation: Moldovyan A. A., Moldovyan D. N., Moldovyan N. A. A novel method for developing post-quantum digital signature algorithms on non-commutative associative algebras. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 44–53. doi:10.31799/1684-8853-2022-1-44-53

Introduction

At present the most widely used digital signature algorithms are based on the computational complexity of the integer factorization problem [1, 2] and the discrete logarithm problem (DLP) [3, 4]. However, both of the said problems can be solved in polynomial time on a quantum computer [5–7]. The expected breakthrough in the technology of quantum computations in the near future makes it extremely urgent to develop practical post-quantum public-key signature algorithms (post-quantum are called algorithms that are resistant to attacks using quantum computers) [8]. Computationally difficult problems other than factorization problem and DLP are to be used as the base cryptographic primitive of post-quantum digital signature algorithms.

In the current field of development of public-key post-quantum cryptoschemes, considerable attention of the cryptographers is paid to the development of cryptoschemes on algebras [9, 10], on Boolean functions [11, 12], and on linear codes [13, 14].

A landmark event in the area of post-quantum cryptography is the worldwide algorithm competition announced by the US National Institute of Standards and Technology (NIST) for the period 2017–2024 with the aim of developing post-quantum standards for digital signature algorithms and public key agreement algorithms. The first round [15] of the competition ended with the selection of 10 signature algorithms and 16 public key agreement algorithms (i. e. 26 public-key algorithms out of 69 initially submitted for participation in the competition) as potential candidates for post-quantum standards. The second round [16] ended with the selection of three signature algorithms and four public key-agreement algorithms (called finalists) for the in-depth analysis in the third round. In addition, three alternative signature algorithms and five alternative key agreement algorithms were selected for consideration at the fourth round of the

competition. For the first time in the NIST cryptographic competitions, along with the finalists, alternative cryptoschemes were selected for final consideration.

However, the most interesting thing is that according to the results of the third round of the competition NIST intends to accept new post-quantum signature algorithms for consideration at the fourth round [17]. In a brief overview of the current results of the competition [17], it is noted: "We are most interested in a general purpose digital signature scheme which is not based on structured lattices". Taking into account that algorithms Dilithium and Falcon, which are based on lattices, are considered the most promising for adopting the post-quantum signature standard, one can conclude that NIST remained somewhat dissatisfied with the current results of the competition in the nomination of signature algorithms. Thus, search for new methods, mechanisms, and algebraic supports for the development of practical post-quantum digital signature algorithms is still an urgent task.

One of attractive primitives of the post-quantum signature algorithms is the hidden discrete logarithm problem (HDLP) defined usually in finite non-commutative associative algebras (FNAAs). Earlier, many different forms of the HDLP were proposed to develop signature algorithms on FNAAs [18–20]. The main feature of the HDLP-based signature schemes is the use of the exponentiation operations in hidden commutative groups and computing the signature in the form of two integers. The latter defines possibility to forge signatures in the case of known secret value of the discrete logarithm in a hidden group, for calculation of the public key secret vectors are used as masking multipliers though. Separate HDLP-based algorithms are characterized by using an auxiliary signature element in the form of a vector **S**. In the last type algorithms for eliminating attacks associated with the use of the vector **S** as a fitting parameter, a doubling of the signature verification equation is proposed [20].

In this paper, we propose a new method for developing the signature algorithms including the exponentiation operations in a hidden group, which is characterized in using a vector **S** as a main element of the signature (*e*, **S**) including the randomization integer *e*. The vector **S** is included in the verification equation two or more times. The latter defines computational infeasibility of forging a signature by calculating the value of **S** from the verification equation. At the same time, with the knowledge of secret masking vectors, it is possible to calculate the vector **S** satisfying the verification equation for arbitrary fixed value of the randomizing signature element *e*. Using the proposed method, a new candidate for post-quantum signature algorithm is developed. To provide higher performance a new four-dimensional FNAA set by a sparse basis vector multiplication table (BVMT) is proposed and used as algebraic support of the signature algorithm.

Four-dimensional FNAA used as algebraic support

A vector space of dimension m, which is set over a finite ground field $GF(p)$, with the additionally defined vector multiplication operation that is distributive at the left and at the right relatively the addition operation is called m-dimensional algebra. A vector $\mathbf{A}$ is presented as an ordered set of its coordinates: $\mathbf{A} = (a_0, a_1, ..., a_{m-1})$ or as a sum of its components: $\mathbf{A} = a_0\mathbf{e}_0 + a_1\mathbf{e}_1 + ... + a_{m-1}\mathbf{e}_{m-1}$, where $\mathbf{e}_i$ ($i = 0, 1, ..., m - 1$) are formal basis vectors. If the vector multiplication is non-commutative and associative, then one gets m-dimensional FNAA.

Usually, the multiplication of two vectors $\mathbf{A} = \sum_{i=0}^{m-1} a_i\mathbf{e}_i$ and $\mathbf{B} = \sum_{j=0}^{m-1} b_j\mathbf{e}_j$ is defined by the following formula: $\mathbf{AB} = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} a_i b_j \mathbf{e}_i\mathbf{e}_j$, where the coordinates a_i and b_i are multiplied as elements of the field $GF(p)$ and every the product of two formal basis vectors is to be replaced by an one-component vector indicated in a cell at the intersection of the i-th row and j-th column of so called BVMT. In general for the case $m = 4$, one vector multiplication operation is implemented as executing about 16 multiplications and 12 additions in $GF(p)$. To reduce computational complexity of the vector multiplication we propose a new sparse BVMT shown as Table 1, which defines a four-dimensional FNAA with reduced two times complexity of the vector multiplication.

A left-sided unit $\mathbf{E}_L$ of the said algebra can be computed from the vector equation $\mathbf{XA} = \mathbf{A}$ that can be reduced to the following two independent systems of two linear equations with unknown values of the coordinates of the vector $\mathbf{X} = (x_0, x_1, x_2, x_3)$:

■ ***Table 1***. Multiplication of basis vectors ($\lambda \neq 0$) in the proposed four-dimensional FNAA with global two-sided unit $\mathbf{E} = (0, 1, 1, 0)$

·	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_0$	0	0	$\mathbf{e}_0$	$\lambda\mathbf{e}_1$
$\mathbf{e}_1$	$\mathbf{e}_0$	$\mathbf{e}_1$	0	0
$\mathbf{e}_2$	0	0	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_3$	0	0

$$\begin{cases} a_1x_0 + a_0x_2 = a_0; \\ a_2x_2 + \lambda a_3x_0 = a_2; \end{cases}$$
$$\begin{cases} a_1x_1 + \lambda a_0x_3 = a_1; \\ a_3x_1 + a_2x_3 = a_2. \end{cases}$$

From the last two systems one easily gets $\mathbf{E}_L = (0, 1, 1, 0)$. The right-sided unit $\mathbf{E}_R$ can be computed from the vector equation $\mathbf{AX} = \mathbf{A}$. The latter gives $\mathbf{E}_R = (0, 1, 1, 0) = \mathbf{E}_L$. One can easily see that the vector $\mathbf{E} = (0, 1, 1, 0)$ acts as global two-sided unit in the considered four-dimensional FNAA. One can show that for different fixed values $\mathbf{A}$ the vector equation $\mathbf{AX} = \mathbf{E}$ has a unique solution or has no solutions. In the first case the vector $\mathbf{A}$ is called invertible and in the second case it is called non-invertible. Inverse value of $\mathbf{A}$ is denoted as the vector $\mathbf{A}^{-1}$. Considering the vector equation $\mathbf{AX} = \mathbf{E}$ or equation $\mathbf{XA} = \mathbf{E}$, one can obtain the invertibility (non-invertibility) condition of the vector $\mathbf{A}$:

$$a_1a_2 \neq \lambda a_0a_3 \ (a_1a_2 = \lambda a_0a_3). \qquad (1)$$

Using the formulas (1) it is easy to find the number on non-invertible vectors (equal to $p^3 + p^2 - p$) and then the number Ω of invertible vectors (order of the multiplicative group of the algebra):

$$\Omega = p(p^2 - 1)(p - 1). \qquad (2)$$

The structure of a FNAA from the view point of its decomposition into a set of commutative subalgebras represents significant interest while using it as algebraic support of the HDLP-based signature algorithms [18]. Next section describes the structure of the introduced four-dimensional FNAA.

Structure of the algebra used as algebraic support

To study the structure of the FNAA set by Table 1, we apply the method used earlier in the paper [18]. Consider the set of the vectors $\mathbf{X}$ that are permutable with a fixed vector $\mathbf{A} = (a_0, a_1, a_2, a_3)$. The set of vectors $\mathbf{X} = (x_0, x_1, x_2, x_3)$ can be computed as solution space of the following vector equation:

$$\mathbf{AX} - \mathbf{XA} = (0, 0, 0, 0). \qquad (3)$$

If $\mathbf{X}_1$ and $\mathbf{X}_2$ are two solutions, then $\mathbf{X}_1 \pm \mathbf{X}_2$ and $\mathbf{X}_1\mathbf{X}_2$ are also solutions. One can show that the set of solutions of the equation (3) represents a subalgebra $\Psi_{\mathbf{A}}$. The said vector equation can be reduced to the following system of four linear equations with the unknowns x_0, x_1, x_2, and x_3:

$$\begin{cases} a_1x_0 + a_0x_2 - a_0x_1 - a_2x_0 + \lambda a_3x_0 = 0; \\ a_1x_1 + \lambda a_0x_3 - a_1x_1 - \lambda a_3x_0 = 0; \\ a_1x_2 + \lambda a_3x_0 - a_2x_2 - \lambda a_0x_3 = 0; \\ a_3x_1 + a_2x_3 - a_1x_3 - a_3x_2 = 0. \end{cases} \qquad (4)$$

The system (4) reduces to the following system of three linear equations:

$$\begin{cases} a_3x_0 - a_0x_3 = 0; \\ (a_1 - a_2)x_0 + a_0(x_2 - x_1) = 0; \\ a_3(x_1 - x_2) + (a_2 - a_1)x_3 = 0. \end{cases} \qquad (5)$$

Depending on the vector $\mathbf{A}$ there are possible the following cases.

I. Case $a_0 = a_3 = 0$. The system (5) reduces to the system

$$\begin{cases} (a_1 - a_2)x_0 = 0; \\ (a_1 - a_2)x_3 = 0. \end{cases}$$

If $a_1 \neq a_2$, then the subalgebra $\Psi_{\mathbf{A}}$ includes the set of vectors described by the following formula:

$$\mathbf{X} = (x_0, x_1, x_2, x_3) = (0, d, h, 0), \qquad (6)$$

where $d, h = 0, 1, ..., p - 1$. The set (6) contains $2p - 1$ non-invertible vectors of the forms $(0, 0, h, 0)$ and $(0, d, 0, 0)$ and $(p - 1)^2$ invertible vectors, i. e., the multiplicative group Γ_1 of the $\Psi_{\mathbf{A}}$ subalgebra has order $\Omega_1 = (p - 1)^2$. A generator system of the group Γ_1 includes two vectors or order $p - 1$. Such group is called a group possessing two-dimensional cyclicity. Subalgebras containing a multiplicative group of the Γ_1 type are called subalgebras of the Ψ_1 type.

If $a_1 = a_2$, then every vector of the considered FNAA is permutable with $\mathbf{A}$. Indeed, in this subcase we have a scalar vector $\mathbf{A} = (0, a_1, a_1, 0)$.

II. Case $a_0 \neq 0$; $a_3 = 0$. The system (5) reduces to the system

$$\begin{cases} x_3 = 0; \\ x_2 = x_1 - \dfrac{a_1 - a_2}{a_0}x_0. \end{cases}$$

The set of elements of the subalgebra $\Psi_{\mathbf{A}}$ is described by the following formula

$$\mathbf{X} = \left(d, h, h - \frac{a_1 - a_2}{a_0}d, 0\right), \qquad (7)$$

where $d, h = 0, 1, ..., p - 1$. Taking into account the non-invertibility condition in (1), for the non-invertible vectors contained in (6) one can write

$$h\left(h-\frac{a_1-a_2}{a_0}d\right)=0.$$

For the subcase $a_1 \neq a_2$, from the latter formula one can conclude that the set (7) contains $2p-1$ non-invertible vectors and we have subalgebra of the Ψ_1 type.

For the subcase $a_1 = a_2$, from the non-invertibility condition in (1) we have $h = 0$ and p non-invertible vectors of the form $(d, 0, 0, 0)$. Respectively, the order of multiplicative group of the subalgebra $\Psi_{\mathbf{A}}$ is equal to $\Omega_2 = p^2 - p = p(p-1)$. The multiplicative group is cyclic and is attributed to the Γ_2 type. Subalgebra containing a multiplicative group of the Γ_2 type is attributed to the Ψ_2 type.

III. Case $a_0 = 0$; $a_3 \neq 0$. The system (5) reduces to the system

$$\begin{cases} x_3 = 0; \\ x_2 = x_1 - \dfrac{a_1-a_2}{a_3}x_3. \end{cases}$$

The set of elements of the subalgebra $\Psi_{\mathbf{A}}$ is described by the following formula:

$$\mathbf{X}=\left(0, d, d-h\frac{a_1-a_2}{a_3}, h\right), \tag{8}$$

where d, $h = 0, 1, \ldots, p-1$. Taking into account the non-invertibility condition in (1), for the non-invertible vectors contained in (8) one can write

$$d\left(d-h\frac{a_1-a_2}{a_3}d\right)=0.$$

For the subcase $a_1 \neq a_2$, from the latter formula one can conclude that the set (8) contains $2p-1$ non-invertible vectors and we have subalgebra of the Ψ_1 type.

For the subcase $a_1 = a_2$, from the non-invertibility condition in (1) we have $d = 0$ and p non-invertible vectors of the form $(0, 0, 0, h)$. Respectively, the order of multiplicative group of the subalgebra $\Psi_{\mathbf{A}}$ is equal to $\Omega_2 = p^2 - p = p(p-1)$. The multiplicative group is cyclic and is attributed to the Γ_2 type. Subalgebra containing a multiplicative group of the Γ_2 type is attributed to the Ψ_2 type.

IV. Case $a_0 \neq 0$; $a_3 \neq 0$. The system (5) reduces to the system

$$\begin{cases} x_3 = x_0\dfrac{a_3}{a_0}; \\ x_2 = x_1 - \dfrac{a_1-a_2}{a_0}x_0. \end{cases}$$

The set of all elements of the subalgebra $\Psi_{\mathbf{A}}$ is described by the following formula:

$$\mathbf{X}=\left(d, h, h+\frac{a_2-a_1}{a_0}d, \frac{a_3}{a_0}d\right), \tag{9}$$

where d, $h = 0, 1, \ldots, p-1$. Taking into account the conditions (1), for the non-invertible vectors contained in (9) we have

$$\lambda d^2\frac{a_3}{a_0}=h^2+dh\frac{a_2-a_1}{a_0}. \tag{10}$$

From the quadratic equation (10) one has solution

$$h=\frac{a_1-a_2}{2a_0}d\pm\sqrt{\frac{(a_1-a_2)^2}{4a_0^2}d^2+\lambda d^2\frac{a_3}{a_0}}= \\ =\frac{a_1-a_2\pm\sqrt{\Delta}}{2a_0}d, \tag{11}$$

where

$$\Delta=(a_1-a_2)^2+4\lambda a_0 a_3. \tag{12}$$

Depending on the value Δ we have the following subcases.

IVa. Subcase Δ is quadratic residue modulo p ($\Delta \neq 0$). From (11) one can see that for every value of $d = 1, 2, \ldots, p-1$ we have two different values of h. This gives $2(p-1)$ nonzero non-invertible vectors. Totally, the number of non-invertible vectors is equal to $2p-1$, therefore the set (9) describes subalgebras of the Ψ_1 type containing multiplicative group of Γ_1 type.

IVb. Subcase Δ is quadratic non-residue modulo p ($\Delta \neq 0$). The equation (11) has no solutions and the set (9) contains only one non-invertible vector, namely, the zero vector (0, 0, 0, 0). The order of multiplicative group of the $\Psi_{\mathbf{A}}$ algebra is $\Omega_3 = p^2 - 1$. This group is attributed to the third type denoted as Γ_3. A subalgebra described by formula (9) represents a field that is isomorphic to $GF(p^2)$. Therefore, the groups of the Γ_3 type are cyclic.

IVc. Subcase $\Delta = 0$. From (11) one can see that for every value of $d = 0, 1, \ldots, p-1$ we have exactly one value of h. This gives p non-invertible vectors, therefore, the set (9) describes subalgebras of the Ψ_2 type containing multiplicative group of Γ_2 type, which has order equal to $\Omega_2 = p(p-1)$.

Like it has been shown in [18], one can prove the following formulas:

i) for the number η of different $\Psi_{\mathbf{A}}$ subalgebras: $\eta = p^2 + p + 1$;

ii) for the number η_1 of different subalgebras of the Ψ_1 type:

$$\eta_1=\frac{p(p+1)}{2}; \tag{13}$$

iii) for the number η_2 of different Ψ_2 subalgebras:

$$\eta_2 = p+1; \quad (14)$$

iv) for the number η_3 of different Ψ_3 subalgebras:

$$\eta_3 = \frac{p(p-1)}{2}. \quad (15)$$

The number of commutative groups of the types Γ_1, Γ_2 and Γ_3, in which the group operation is the vector multiplication, is defined by the formulas (13)–(15), correspondingly.

Proposed method

Into the base of the proposed method for development post-quantum digital signature algorithm is put the idea of using the vectors **G** and **H** contained in a hidden group to compute both the public key in the form of several vectors, for example, $\mathbf{Y}_1$, $\mathbf{Z}_1$, $\mathbf{Y}_2$, $\mathbf{Z}_2$ (which are pairwise non-permutable) and the signature element of the form of vector $\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^n\mathbf{H}^r\mathbf{C}^{-1}$, where non-permutable vectors **B** and **C** are masking multipliers. The design of concrete signature scheme should be so that computation of the non-negative integers n and m allows one to get the value of **S**, which satisfies the signature verification equation with several occurrences of the signature element **S** that is non-permutable with every element of the public key. For example, in the case of two occurrences of the vector **S** one can use the verification equation of the following form

$$\mathbf{R} = (\mathbf{Y}_1\mathbf{S}\mathbf{Z}_1)^e (\mathbf{Y}_2\mathbf{S}\mathbf{Z}_2)^{e^2}, \quad (16)$$

where e is the signature randomization element in the form of a natural number computed as a hash function value from an electronic document M (to be signed) and the vector **R**. Including in the signature generation procedure a step of computation of the vector **R** in the form $\mathbf{R} = \mathbf{A}\mathbf{G}^k\mathbf{H}^t\mathbf{A}^{-1}$ provides potential possibility of finding the required values of the vector **S**.

To implement this method one needs to use a FNAA containing sufficiently large number of commutative groups. The proposed four-dimensional FNAA suits well as algabraic support of the method. Using the results on study its structure one can propose algorithms for generation of the vectors **G** and **H** defining the type of the hidden group (Γ_1, Γ_2, or Γ_3). In accordance with the formulas (13), (14), and (15), it appears that the most attractive is the use of hidden groups of the types Γ_1 and Γ_3. In the next section we describe a signature scheme in which the hidden group of the Γ_1 type is used. However, the number of the Γ_2 groups is also sufficiently large, therefore the use of a hidden group of the Γ_2 type seems to be not critical from the security point of view. Besides, there is no need to fix the hidden group type and the user can select it at stage of generating the public key.

Proposed candidate for post-quantum signature scheme

Suppose that the four-dimensional FNAA is defined over the field $GF(p)$ with prime characteristic $p = 2q + 1$, where q is a 256-bit prime. It is easy to generate such primes, including the case, when the structure of primes q and p is such that the multiplication modulo p and modulo q can be executed without using the arithmetic division operation (this item has practical significance to get significantly higher performance of the digital signature algorithm described below). In the developed signature scheme a hidden group of the Γ_1 type is used as a hidden group. To set the latter the following algorithm for generating its minimum generator system $\langle \mathbf{G}, \mathbf{H} \rangle$ is used.

1. Select at random an invertible vector $\mathbf{A} = (a_0, a_1, a_2, a_3)$ such that $a_1a_3 \neq \lambda a_0a_3$ and, using the formula (12), compute the value of Δ.

2. If $\Delta \neq 0$ is a quadratic non-residue, then go to step 1. Otherwise set integer variable $d = 1$.

3. Using the formula (11), compute the integer h.

4. Using the formula (9), compute the vector $\mathbf{X} = (x_0, x_1, x_2, x_3)$.

5. If $a_1a_3 = \lambda a_0a_3$, then set the variable $d \leftarrow d + 1$ and go to step 3. Otherwise compute the vector $\mathbf{H} = \mathbf{X}^{(p-1)/q} = \mathbf{X}^2 = (h_0, h_1, h_2, h_3)$.

6. If $\mathbf{H} = \mathbf{E}$ or $(h_0, h_3) = (0, 0)$, then set the variable $d \leftarrow d + 1$ and go to step 3. Otherwise generate a primitive element $\alpha \in GF(p)$ and compute the scalar vector $\mathbf{L} = (0, \alpha, \alpha, 0)$.

7. Generate a random integer $k < q$ and compute the vector $\mathbf{G} = \mathbf{L}^k\mathbf{H}$. Then output the vectors **H** and **G**.

In line with the proposed method, the following procedure of computing the public key is proposed.

Algorithm for computation of the public key.

1. Generate private vectors **G** and **H** that compose a minimum generator system $\langle \mathbf{G}, \mathbf{H} \rangle$ of a of hidden group of the Γ_1 type (i. e. a primary group of order q^2, which possesses two-dimensional cyclicity).

2. Generate at random invertible vectors **A**, **B**, and **C** that are pairwise non-permutable, every of which in also non-permutable with each of the vectors **G** and **H**.

3. Generate uniformly random integers $u < q$ and $w < q$. Then compute the following four vectors serving as elements of the public key ($\mathbf{Y}_1$, $\mathbf{Z}_1$, $\mathbf{Y}_2$, $\mathbf{Z}_2$):

$$\mathbf{Y}_1 = \mathbf{AG}^u\mathbf{B};\ \mathbf{Z}_1 = \mathbf{CHA}^{-1};$$

$$\mathbf{Y}_2 = \mathbf{AH}^w\mathbf{B};\ \mathbf{Z}_1 = \mathbf{CGA}^{-1}. \quad (17)$$

(Calculation of the vector $\mathbf{A}^{-1}$ can be executed as finding solution of the vector equation $\mathbf{AX} = \mathbf{E}$.)

The size of the public key is equal approximately to 4096 bits (512 bytes). The private key is the following set of values: u, w, **G**, **H**, **A**, **B**, and **C**. The size of the private key is equal approximately to 5632 bits (704 bytes).

Signature generation algorithm.

Suppose the owner of the public key wishes to sign an electronic document M. Then he can use the following algorithm.

1. Generate uniformly random integers $k < q$ and $t < q$ and compute the vector **R**:

$$\mathbf{R} = \mathbf{AG}^k\mathbf{H}^t\mathbf{A}^{-1}. \quad (18)$$

2. Using a pre-agreed hash function f, compute the first signature element $e = f(M, \mathbf{R})$.

3. Calculate the integers n and r as follows:

$$n = \frac{k - ue - e^2}{e + e^2} \bmod q;$$

$$r = \frac{t - we^2 - e}{e + e^2} \bmod q.$$

4. Calculate the second signature element in the form of vector **S**:

$$\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^n\mathbf{H}^r\mathbf{C}^{-1}. \quad (19)$$

The size of the signature $(e, \mathbf{S})$ is equal approximately to 1280 bits (160 bytes). Computational complexity of the signature generation algorithm can be estimated as 4 exponentiation operations in the FNAA set by Table 1 (≈12 288 multiplications in $GF(p)$).

Signature verification algorithm.

To verify the signature $(e, \mathbf{S})$ assigned to document M one can use the following procedure.

1. Compute the vector **R***:

$$\mathbf{R}^* = \left(\mathbf{Y}_1\mathbf{SZ}_1\right)^e\left(\mathbf{Y}_2\mathbf{SZ}_2\right)^{e^2}. \quad (20)$$

2. Using a pre-agreed hash function f, compute the value $e^* = f(M, \mathbf{R}^*)$.

3. Compare the values e^* and e. If $e^* = e$, then the signature $(e, \mathbf{S})$ is accepted as genuine. Otherwise the signature is rejected.

Computational complexity of the signature verification algorithm can be estimated as 2 exponentiations in the FNAA used as algebraic support (≈6144 multiplications modulo p).

Proof of the signature scheme correctness.

Consider a signature $(e, \mathbf{S})$ to document M, which is computed correctly in full correspondence with the signature generation procedure, while using the correct signer's private key. In line with the signature verification algorithm, for the signature $(e, \mathbf{S})$ one can write the following:

$$\mathbf{R}^* = \left(\mathbf{Y}_1\mathbf{SZ}_1\right)^e\left(\mathbf{Y}_2\mathbf{SZ}_2\right)^{e^2} =$$

$$= \left(\mathbf{AG}^u\mathbf{BB}^{-1}\mathbf{G}^n\mathbf{H}^r\mathbf{C}^{-1}\mathbf{CHA}^{-1}\right)^e \times$$

$$\times \left(\mathbf{AH}^w\mathbf{BB}^{-1}\mathbf{G}^n\mathbf{H}^r\mathbf{C}^{-1}\mathbf{CGA}^{-1}\right)^{e^2} =$$

$$= \left(\mathbf{AG}^{u+n}\mathbf{H}^{r+1}\mathbf{A}^{-1}\right)^e\left(\mathbf{AH}^{w+r}\mathbf{G}^{n+1}\mathbf{A}^{-1}\right)^{e^2} =$$

$$= \left(\mathbf{AG}^{(u+n)e}\mathbf{H}^{(r+1)e}\mathbf{A}^{-1}\right)\left(\mathbf{AH}^{(w+r)e^2}\mathbf{G}^{(n+1)e^2}\mathbf{A}^{-1}\right) =$$

$$= \mathbf{AG}^{(u+n)e+(n+1)e^2}\mathbf{H}^{(r+1)e+(w+r)e^2}\mathbf{A}^{-1} =$$

$$= \mathbf{AG}^{n\left(e+e^2\right)+eu+e^2}\mathbf{H}^{r\left(e+e^2\right)+we^2+e}\mathbf{A}^{-1} =$$

$$= \mathbf{AG}^{\frac{k-eu-e^2}{e+e^2}\left(e+e^2\right)+eu+e^2}\mathbf{H}^{\frac{k-we^2-e}{e+e^2}\left(e+e^2\right)+we^2+e}\mathbf{A}^{-1} =$$

$$= \mathbf{AG}^k\mathbf{H}^t\mathbf{A}^{-1} = \mathbf{R} \Rightarrow$$

$$\Rightarrow f\left(M, \mathbf{R}^*\right) = f\left(M, \mathbf{R}\right) \Rightarrow e^* = e.$$

The final equality means the input signature passes the verification procedure as a genuine signature, i. e., the signature scheme performs correctly.

Discussion

The developed signature algorithm uses the exponentiation operation in a hidden commutative group and powers of these operations are secret, like in the known HDLP-based signature schemes [18–20]. However, in the latter schemes for computing a signature it is sufficient to use only the values of the said powers, while in the signature scheme described in the previous section, without using the secret vectors **G**, **H**, **A**, **B**, and **C** a valid signature cannot be directly generated. This is due to a new mechanism used for masking the hidden group, which is presented by formulas (17).

If a potential forger knows the powers u and w and the minimum generator system of the hidden group $\langle \mathbf{G}, \mathbf{H} \rangle$, then he will be able to forge signatures as follows:

1. Compute the vectors $\mathbf{U} = \mathbf{G}^u$ and $\mathbf{W} = \mathbf{H}^w$.

2. Using the public key elements and considering the vectors $\mathbf{A}'$, $\mathbf{B}'^{-1}$, and $\mathbf{C}'$ as unknowns compose the following system of four linear vector equations:

$$\begin{cases} \mathbf{Y}_1\mathbf{B}'^{-1} = \mathbf{A}'\mathbf{U}; \\ \mathbf{Z}_1\mathbf{A}' = \mathbf{C}'\mathbf{H}; \\ \mathbf{Y}_2\mathbf{B}'^{-1} = \mathbf{A}'\mathbf{W}; \\ \mathbf{Z}_2\mathbf{A}' = \mathbf{C}'\mathbf{G}. \end{cases} \quad (21)$$

[The system (21) reduces to a system of 16 linear equations in $GF(p)$ with 12 unknown coordinates of the vectors $\mathbf{A}'$, $\mathbf{B}'^{-1}$, and $\mathbf{C}'$. Evidently, the system (21) has a solution, namely, $\mathbf{A}' = \mathbf{A}$, $\mathbf{B}'^{-1} = \mathbf{B}^{-1}$, and $\mathbf{C}' = \mathbf{C}$.]

3. Solve the system (21). Then, using the found values of $\mathbf{A}$, $\mathbf{B}^{-1}$, $\mathbf{C}$ and signature generation procedure from previous section, generate a signature. [If the system (21) has a solution different from $(\mathbf{A}', \mathbf{B}'^{-1}, \mathbf{C}') = (\mathbf{A}, \mathbf{B}^{-1}, \mathbf{C})$, then it will also provide generation of a valid signature.]

Currently, the proposed method and the algorithm of the case is quite new and is still unclear what way the signature scheme can be efficiently attached, when no element of the private key is known for the attacker. One can propose the next general approach for forging a signature, which consists in finding an alternative representation of the public key, i. e., finding the values $\mathbf{A}'$, $\mathbf{B}'$, $\mathbf{C}'$, $\mathbf{G}'$, $\mathbf{G}_w$, $\mathbf{H}'$, $\mathbf{H}_w$, where $\mathbf{G}'$, $\mathbf{G}_w$, $\mathbf{H}'$, and $\mathbf{H}_w$ are pairwise permutable vectors, such that they satisfy the following system of the vector equations

$$\begin{cases} \mathbf{Y}_1\mathbf{B}'^{-1} = \mathbf{A}'\mathbf{G}_u; \\ \mathbf{Z}_1\mathbf{A}' = \mathbf{C}'\mathbf{H}'; \\ \mathbf{Y}_2\mathbf{B}'^{-1} = \mathbf{A}'\mathbf{H}_w; \\ \mathbf{Z}_2\mathbf{A}' = \mathbf{C}'\mathbf{G}'. \end{cases} \quad (22)$$

One can easily show that for such alternative representation of the public key a valid signature can be calculated using a signature verification algorithm which is similar to that described in previous section. However, all of the equations in (22) contain products of a couple of unknowns, therefore, solving the system (22) appears to be a computationally hard problem.

Indeed, the requirement of permutability on the unknown vectors $\mathbf{G}'$, $\mathbf{G}_w$, $\mathbf{H}'$, and $\mathbf{H}_w$ adds three vector equations to (22) and one gets the system of seven equations in the FNAA, which reduces to the system of 28 quadratic equations in $GF(p)$ with 28 unknowns. Finding a solution for such systems is a computationally difficult problem [21, 22]. One can suppose that the computational complexity of finding a solution of the system (22) defines the security level of the proposed signature scheme.

Development of the methods for solving the system (22) and estimation of their computational complexity is an independent research task. We would only like to note that improving the complexity of the solution of the mentioned computational problem can be achieved by increasing the size of the prime p and/or increasing the public key size. The latter can be implemented by calculating two additional public-key elements $\mathbf{Y}_3$ and $\mathbf{Z}_3$, using additional private integers $b < q$ and $d < q$, which are also uniformly random values: $\mathbf{Y}_3 = \mathbf{A}\mathbf{H}^b\mathbf{B}$; $\mathbf{Z}_3 = \mathbf{C}\mathbf{G}^d\mathbf{A}^{-1}$. Respectively, such modification of the public key requires updating the verification equation. For example the following one can be used:

$$\mathbf{R} = \left(\mathbf{Y}_1\mathbf{S}\mathbf{Z}_1\right)^e \left(\mathbf{Y}_2\mathbf{S}\mathbf{Z}_2\right)^{e^2} \left(\mathbf{Y}_3\mathbf{S}\mathbf{Z}_3\right)^{e+e^2}. \quad (23)$$

(In line with the method presented in Section "Proposed method" and signature algorithm described in previous Section "Proposed candidate for post-quantum signature scheme", the reader can easily update the signature generation procedure in correspondence with the modified versions of the public key and verification equation.)

The use of a hypothetical quantum computer is not effective for solving the specified problem of solving the system of equations (22). In addition, when analyzing the security of the proposed digital signature algorithm, there is no need to solve DLP or HDLP, despite the fact that the exponential operations play a significant role in the developed algorithm. For example, in contrast to the known

■ ***Table 2***. Comparison with some known post-quantum digital signature algorithms

Signature scheme	Signature size, byte	Public key size, byte	Signature generation rate, arb. un.	Signature verification rate, arb. un.
Falcon [24]	1280	1793	50	25
Dilithium [25]	2701	1472	15	2
HDLP-based [19]	192	768	50	80
HDLP-based [20]	192	512	40	80
HDLP-based [26]	96	576	30	40
Proposed	160	512	140	290

HDLP-based algorithms in which the exponentiations plays a fundamental role, the proposed algorithm can be modified in such a way that, when generating a public key, exponentiation operations will not be used.

Thus, in comparison with the known HDLP-based algorithms, the introduced method and the developed digital signature algorithm is characterized in that the appearance of computationally efficient algorithms for solving DLP and HLP does not mean that the signature algorithm has ceased to be safe. In this connection one can notice that the presence of a large-sized prime divisor in the decomposition of the order of the multiplicative group of the $GF(p)$ field is not a critical requirement. This feature simplifies the implementation of the algorithm when using the FNAA, set over a field $GF(2^s)$, as algebraic support.

A draft comparison of the developed signature algorithm with two finalists (Falcon and Dilithium) of the NIST competition [23] and some of the HDLP-based signature schemes are presented in Table 2. The algorithm proposed in this article has a significant advantage in the sizes of the signature and public key. Besides, it has higher performance.

Conclusion

Like in a number of known HDLP-based signature schemes, in the developed algorithm a hidden group is used, but the latter algorithm more precisely should be called an algorithm with a hidden group. The proposed method can be used to develop many different algorithms with a hidden group, which are attractive as candidates for practical post-quantum signature algorithms.

The results of this article can be considered as a starting point for the formation of a new concept of the development of post-quantum digital signature algorithms on non-commutative algebras, in framework of which one will be able potentially to reduce significantly the size of the public key and the signature while simultaneous increasing performance.

Financial support

This research is partially supported by RFBR (project No 21-57-54001-Вьет_а) and budget theme No FFZF-2022-0007.

References

1. Rivest R. L., Shamir A., Adleman L. M. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 1978, vol. 21, pp. 120–126.
2. Chiou S. Y. Novel digital signature schemes based on factoring and discrete logarithms. *International Journal of Security and Its Applications*, 2016. vol. 10, no. 3, pp. 295–310.
3. ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 1985, vol. IT-31, no. 4, pp. 469–472.
4. Schnorr C. P. Efficient signature generation by smart cards. *Journal of Cryptology*, 1991, vol. 4, pp. 161–174.
5. Shor P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer. *SIAM Journal of Computing*, 1997, vol. 26, pp. 1484–1509.
6. Ekert A., Jozsa R. Quantum computation and Shor's factoring algorithm. *Reviews of Modern Physics*, 1996, vol. 68, pp. 733–752.
7. Smolin J. A., Smith G., Vargo A. Oversimplifying quantum factoring. *Nature*, 2013, vol. 499, no. 7457, pp. 163–165.
8. *Federal Register. Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms*. Available at: https://www.gpo.gov/fdsys/pkg/FR-2016-12-20/pdf/2016-30615.pdf (accessed 24 November 2021).
9. Kuzmin A. S., Markov V. T., Mikhalev A. A., Mikhalev A. V., Nechaev A. A. Cryptographic algorithms on groups and algebras. *Journal of Mathematical Sciences*, 2017, vol. 223, no. 5, pp. 629–641.
10. Moldovyan D. N. Post-quantum public key-agreement scheme based on a new form of the hidden logarithm problem. *Computer Science Journal of Moldova*, 2019, vol. 27, no. 1(79), pp. 56–72.
11. Agibalov G. P., Pankratova I. A. Asymmetric cryptosystems on Boolean functions. *Prikl. Diskr. Mat.*, 2018, no. 40, pp. 23–33. doi:10.17223/20710410/40/3
12. Agibalov G. P. ElGamal cryptosystems on Boolean functions. *Prikl. Diskr. Mat.*, 2018, no. 42, pp. 57–65. doi:10.17223/20710410/42/4
13. Alamelou Q., Blazy O., Cauchie S., Gaborit Ph. A code-based group signature scheme. *Designs, Codes and Cryptography*, 2017, vol. 82, no. 1–2, pp. 469–493.
14. Kosolapov Y. V., Turchenko O. Y. On the construction of a semantically secure modification of the McEliece cryptosystem. *Prikl. Diskr. Mat.*, 2019, no. 45, pp. 33–43. doi:10.17223/20710410/45/4
15. Alagic G., Alperin-Sheriff J., Apon D., Cooper D., Dang Q., Miller C., Moody D., Peralta R., Perlner R., Robinson A., Smith-Tone D., and Liu Y. *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process*. Ser. NIST Interagency/Internal Report (NISTIR). National Institute of Standards and Technology, Gaithersburg, MD, 2019. https://doi.org/10.6028/NIST.IR.8240. Available at: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927303 (accessed 24 November 2021).

16. Moody D., Alagic G., Apon D., Cooper D., Dang Q., Kelsey J., Liu Y., Miller C., Peralta R., Perlner R., Robinson A., Smith-Tone D., and Alperin-Sheriff J. *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process*. Ser. NIST Interagency/Internal Report (NISTIR). National Institute of Standards and Technology, Gaithersburg, MD, 2020. Available at: https://doi.org/10.6028/NIST.IR.8309 (accessed 24 November 2021).
17. Moody D. *NIST Status Update on the 3rd Round*. National Institute of Standards and Technology, Gaithersburg, MD, 2020. Available at: https://csrc.nist.gov/CSRC/media/Presentations/status-update-on-the-3rd-round/images-media/session-1-moody-nist-round-3-update.pdf (accessed 24 November 2021).
18. Moldovyan D. N. A practical digital signature scheme based on the hidden logarithm problem. *Computer Science Journal of Moldova*, 2021, vol. 29, no. 2(86), pp. 206–226.
19. Moldovyan N. A., Moldovyan A. A. Candidate for practical post-quantum signature scheme. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2020, vol. 16, iss. 4, pp. 455–461. doi: 10.21638/11701/spbu10.2020.410
20. Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. A novel method for development of post-quantum digital signature schemes. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 6 pp. 21–29. doi:10.31799/1684-8853-2020-6-21-29
21. Shuaiting Q., Wenbao H., Yifa Li, Luyao J. Construction of extended multivariate public key cryptosystems. *International Journal of Network Security*, 2016, vol. 18, no. 1, pp. 60–67.
22. Jintai D., Dieter S. *Multivariable Public Key Cryptosystems*. 2004. Available at: https://eprint.iacr.org/2004/350.pdf (accessed 24 November 2021).
23. *Round 3 Finalists: Public-key Encryption and Key-establishment Algorithms*. Available at: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions (accessed 24 November 2021).
24. *Fast-Fourier lattice-based compact signatures over NTRU*. Available at: https://falcon-sign.info/ (accessed 24 November 2021).
25. Ducas L., Kiltz E., Lepoint T., Lyubashevsky V., Schwabe P., Seiler G., Stehlé D. *CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme*. https://eprint.iacr.org/2017/633.pdf Available at: https://pq-crystals.org/dilithium/index.shtml (accessed 24 November 2021).
26. Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. A new design of the signature schemes based on the hidden discrete logarithm problem. *Quasigroups and Related Systems*, 2021, vol. 29, no. 1, pp. 97–106.

УДК 003.26
doi:10.31799/1684-8853-2022-1-44-53

Новый способ построения постквантовых алгоритмов цифровой подписи на некоммутативных алгебрах

А. А. Молдовян[a], доктор техн. наук, главный научный сотрудник, orcid.org/0000-0001-5480-6016, maa1305@yandex.ru
Д. Н. Молдовян[a], канд. техн. наук, научный сотрудник, orcid.org/0000-0001-5039-7198
Н. А. Молдовян[a], доктор техн. наук, главный научный сотрудник, orcid.org/0000-0002-4483-5048
[a]Санкт-Петербургский Федеральный исследовательский центр РАН, 14-я линия В. О., 39, Санкт-Петербург, 199178, РФ

Введение: разработка практичных постквантовых схем подписи является одним из текущих вызовов в области криптографии. Недавно предложены несколько кандидатов на постквантовые схемы цифровой подписи, в которых используется операция экспоненцирования в скрытой группе, содержащейся в некоммутативной алгебре. Поиск новых механизмов использования скрытой группы при разработке схем цифровой подписи, стойких к квантовым атакам, представляет существенный практический интерес. **Цель:** разработать новый способ построения постквантовых алгоритмов цифровой подписи на конечных некоммутативных ассоциативных алгебрах. **Результаты:** предложены новый способ разработки алгоритмов цифровой подписи на некоммутативных алгебрах и новая четырехмерная некоммутативная алгебра, заданная над простым полем $GF(p)$, в качестве носителя указанных алгоритмов. Благодаря заданию операции векторного умножения по прореженным таблицам умножения базисных векторов обеспечивается повышение производительности алгоритмов. Изучение строения алгебры показало, что она представима в виде множества коммутативных подалгебр, попарно пересекающихся строго в множестве всех скалярных векторов. Предложенный метод отличается использованием одного из элементов подписи $(e, \mathbf{S})$ в виде вектора $\mathbf{S}$, вычисляемого как замаскированное произведение степеней двух элементов $\mathbf{G}$ и $\mathbf{H}$ скрытой коммутативной группы: $\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{r}\mathbf{C}^{-1}$, где неперестановочные векторы $\mathbf{B}$ и $\mathbf{C}$ являются маскирующими множителями; натуральные числа n и r вычисляются в зависимости от подписываемого документа M и открытого ключа. Пара $<\mathbf{G}, \mathbf{H}>$ составляет базис скрытой группы. Уравнение верификации подписи имеет вид $\mathbf{R} = (\mathbf{Y}_1\mathbf{S}\mathbf{Z}_1)^{e}(\mathbf{Y}_2\mathbf{S}\mathbf{Z}_2)^{e^2}$, где попарно неперестановочные векторы $\mathbf{Y}_1$, $\mathbf{Z}_1$, $\mathbf{Y}_2$ и $\mathbf{Z}_2$ являются элементами открытого ключа; натуральное число e вычисляется в зависимости от значения M и вектора $\mathbf{R}$. **Практическая значимость:** благодаря достаточно малым размерам подписи и открытого ключа и высокой производительности разработанная схема цифровой подписи представляет интерес как практичный постквантовый алгоритм подписи. Предложенный метод может быть использован для разработки стандарта на постквантовый алгоритм цифровой подписи.

Ключевые слова — постквантовые криптосхемы, компьютерная безопасность, электронная цифровая подпись, задача дискретного логарифмирования, конечные некоммутативные алгебры, ассоциативные алгебры, циклические группы, многомерная цикличность.

Для цитирования: Moldovyan A. A., Moldovyan D. N., Moldovyan N. A. A novel method for developing post-quantum digital signature algorithms on non-commutative associative algebras. *Информационно-управляющие системы*, 2022, № 1, с. 44–53. doi:10.31799/1684-8853-2022-1-44-53
For citation: Moldovyan A. A., Moldovyan D. N., Moldovyan N. A. A novel method for developing post-quantum digital signature algorithms on non-commutative associative algebras. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 44–53. doi:10.31799/1684-8853-2022-1-44-53

Финансовая поддержка
Исследование частично поддержано РФФИ (проект № 21-57-54001-Вьет_а) и бюджетной темой № FFZF-2022-0007.

Уважаемые авторы!

При подготовке рукописей статей необходимо руководствоваться следующими рекомендациями.

Статьи должны содержать изложение новых научных результатов. Название статьи должно быть кратким, но информативным. В названии недопустимо использование сокращений, кроме самых общепринятых (РАН, РФ, САПР и т. п.).

Текст рукописи должен быть оригинальным, а цитирование и самоцитирование корректно оформлено.

Объем статьи (текст, таблицы, иллюстрации и библиография) не должен превышать эквивалента в 20 страниц, напечатанных на бумаге формата А4 на одной стороне через 1,5 интервала Word шрифтом Times New Roman размером 13, поля не менее двух сантиметров.

Обязательными элементами оформления статьи являются: индекс УДК, заглавие, инициалы и фамилия автора (авторов), ученая степень, звание (при отсутствии — должность), полное название организации, аннотация и ключевые слова на русском и английском языках, ORCID и электронный адрес одного из авторов. При написании аннотации не используйте аббревиатур и не делайте ссылок на источники в списке литературы. Предоставляйте подрисуночные подписи и названия таблиц на русском и английском языках.

Статьи авторов, не имеющих ученой степени, рекомендуется публиковать в соавторстве с научным руководителем, наличие подписи научного руководителя на рукописи обязательно; в случае самостоятельной публикации обязательно предоставляйте заверенную по месту работы рекомендацию научного руководителя с указанием его фамилии, имени, отчества, места работы, должности, ученого звания, ученой степени.

Простые **формулы** набирайте в Word, сложные с помощью редактора Mathtype или Equation. Для набора одной формулы не используйте два редактора; при наборе формул в формульном редакторе знаки препинания, ограничивающие формулу, набирайте вместе с формулой; для установки размера шрифта в Mathtype никогда не пользуйтесь вкладкой Other, Smaller, Larger, используйте заводские установки редактора, не подгоняйте размер символов в формулах под размер шрифта в тексте статьи, не растягивайте и не сжимайте мышью формулы, вставленные в текст; пробелы в формуле ставьте только после запятых при перечислении с помощью Ctrl+Shift+Space (пробел); не отделяйте пробелами знаки: + = − ×, а также пространство внутри скобок; для выделения греческих символов в Mathtype полужирным начертанием используйте Style → Other → bold.

Для набора формул в Word никогда не используйте вкладки: «Уравнение», «Конструктор», «Формула» (на верхней панели: «Вставка» — «Уравнение»), так как этот ресурс предназначен только для внутреннего использования в Word и не поддерживается программами, предназначенными для изготовления оригинал-макета журнала.

При наборе символов в тексте помните, что символы, обозначаемые латинскими буквами, набираются светлым курсивом, русскими и греческими — светлым прямым, векторы и матрицы — прямым полужирным шрифтом.

Подробнее см. pdf-файл «Правила подготовки рукописей» (стр. 11) на сайте https://guap.ru/ric

Иллюстрации:

— рисунки, графики, диаграммы, блок-схемы предоставляйте в виде отдельных исходных файлов, поддающихся редактированию, используя векторные программы: Visio (*.vsd, *.vsdx); Adobe Illustrator (*.ai); Coreldraw (*.cdr, версия не выше 15); Excel (*.xls); Word (*.docx); AutoCad, Matlab (экспорт в PDF, EPS, SVG, WMF, EMF); Компас (экспорт в PDF), веб-портал DRAW.IO (экспорт в PDF);

— фото и растровые — в формате *.tif, *.png с максимальным разрешением (не менее 300 pixels/inch).

Наличие подрисуночных подписей и названий таблиц на русском и английском языках обязательно (желательно не повторяющих дословно комментарии к рисункам в тексте статьи).

В редакцию предоставляются:

— сведения об авторе (фамилия, имя, отчество, место работы, должность, ученое звание, учебное заведение и год его окончания, ученая степень и год защиты диссертации, область научных интересов, количество научных публикаций, домашний и служебный адреса и телефоны, e-mail), фото авторов: анфас, в темной одежде на белом фоне, должны быть видны плечи и грудь, высокая степень четкости изображения без теней и отблесков на лице, фото можно представить в электронном виде в формате *.tif, *.png, *.jpg с максимальным разрешением — не менее 300 pixels/inch при минимальном размере фото 40×55 мм;

— экспертное заключение.

Список литературы составляется по порядку ссылок в тексте и оформляется следующим образом:

— для книг и сборников — фамилия и инициалы авторов, полное название книги (сборника), город, издательство, год, общее количество страниц, doi;

— для журнальных статей — фамилия и инициалы авторов, полное название статьи, название журнала, год издания, номер журнала, номера страниц, doi;

— ссылки на иностранную литературу следует давать на языке оригинала без сокращений;

— при использовании web-материалов указывайте адрес сайта и дату обращения.

Список литературы оформляйте двумя отдельными блоками по образцам lit.dot на сайте журнала (http://i-us.ru/paperrules): Литература и References.

Более подробно правила подготовки текста с образцами изложены на нашем сайте в разделе «Руководство для авторов».

Контакты

Куда: 190000, Санкт-Петербург,
Б. Морская ул., д. 67, ГУАП, РИЦ
Кому: Редакция журнала «Информационно-управляющие системы»
Тел.: (812) 494-70-02
Эл. почта: ius.spb@gmail.com
Сайт: www.i-us.ru

УДК 004; 621.398
doi:10.31799/1684-8853-2022-1-54-67

Научные статьи
Articles

Метод обеспечения информационной безопасности сети VoIP-телефонии с прогнозом стратегии вторжений нарушителя

В. А. Липатников[a], *доктор техн. наук, профессор, orcid.org/0000-0002-3736-4743, lipatnikovanl@mail.ru*
А. А. Шевченко[a], *научный сотрудник, orcid.org/0000-0001-9113-1089*
В. С. Косолапов[a], *адъюнкт, orcid.org/0000-0001-8464-779X*
Д. С. Сокол[a], *оператор роты (научной), orcid.org/0000-0002-1532-8872*
[a]*Военная академия связи им. Маршала Советского Союза С. М. Буденного, Тихорецкий пр., 3, Санкт-Петербург, 194064, РФ*

Введение: *развитие технологий в сфере информации и телекоммуникации, а также унификация сетей и, в частности, построение распределенных сетей VoIP-телефонии позволяют сформулировать проблему, заключающуюся в том, что известные методы управления защитой VoIP сетей в современных условиях недостаточно эффективны, так как учитывают только одну сторону информационного противоборства.* ***Цель:*** *разработать метод обеспечения информационной безопасности сети VoIP-телефонии, позволяющий повысить вероятность защищенности VoIP сети путем уменьшения затрачиваемого времени, необходимого для анализа действий нарушителя, анализа и обработки рисков в условиях воздействия нарушителя.* ***Результаты:*** *на основе предложенной структуры системы управления информационной безопасностью, интегрируемой в VoIP сеть, разработан метод обеспечения информационной безопасности сети VoIP-телефонии в условиях воздействия нарушителя за счет внедрения процессов поддержки принятия решения в системе управления информационной безопасностью VoIP сети с использованием распределенных по сегментам интеллектуальных средств обнаружения вторжений. Данный метод позволил построить граф событий действий нарушителя, на основе которого проведено математическое моделирование атак MiTM и SPIT на сеть VoIP-телефонии. В результате моделирования получена зависимость успешного воздействия от внутренних и внешних характеристик атак, которая является основой разработанного программного обеспечения, позволяющего получить значения вероятности защищенности VoIP сети от степени воздействия нарушителя для дальнейшего выбора адекватных мер по управлению информационной безопасностью сети VoIP-телефонии. Метод включает в себя процессы анализа цифрового потока и определения параметров протоколов и профилей атак нарушителей.* ***Практическая значимость:*** *разработанный метод предоставляет возможность исследовать вопросы по защищенности сети VoIP-телефонии, на которую проводится воздействие со стороны нарушителей.*

Ключевые слова *— VoIP сеть, информационная безопасность, MiTM, SPIT, марковский случайный процесс, модель атаки.*

Для цитирования: Липатников В. А., Шевченко А. А., Косолапов В. С., Сокол Д. С. Метод обеспечения информационной безопасности сети VoIP-телефонии с прогнозом стратегии вторжений нарушителя. *Информационно-управляющие системы*, 2022, № 1, с. 54–67. doi:10.31799/1684-8853-2022-1-54-67

For citation: Lipatnikov V. A., Shevchenko A. A., Kosolapov V. S., Sokol D. S. Method for ensuring information security of a VoIP telephony network with a forecast of an intruder's intrusion strategy. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 54–67 (In Russian). doi:10.31799/1684-8853-2022-1-54-67

Введение

Развитие технологий в сфере информации и телекоммуникации, а также унификация сетей и, в частности, построение распределенных сетей VoIP-телефонии повлияло на число исследований проблем, связанных с обеспечением информационной безопасности [1].

Ужесточение требований к средствам защиты информации (СЗИ) влечет за собой более детальное исследование вопросов информационной безопасности (ИБ). Разрабатываются новые методы обеспечения ИБ. Одним из них является способ, позволяющий формировать управляющие воздействия на СЗИ и объекты сети на основе анализа информации о нарушителе, полученной с помощью ложной инфраструктуры [2]. Существуют способы, которые позволяют управлять сетевой безопасностью за счет фиксации уязвимостей сети одновременно с ее функционированием [3, 4]. В вышеперечисленных решениях управление обеспечением ИБ носит реактивный характер, не учитывается прогнозирование воздействий на сеть.

Вывод по обзору релевантных работ [2–4] заключается в том, что при разработке методов ИБ недостаточно уделено внимание анализу динамики действий нарушителя. Одним из требований, которые предъявляются к информационно-вычислительным системам (ИВС), включая и VoIP сети, является реализация способа находить аномалии и возможные вторжения со стороны нарушителей в реальном времени. Возникает противоречие между современными и развивающимися возможностями нарушителей по вторжениям в сети и существующими способами защиты VoIP сети.

В связи с этим результаты анализа релевантных работ позволяют утверждать, что задача защиты инфраструктуры VoIP сети от вторжений со стороны внешних и внутренних нарушителей актуальна.

Целью исследования является повышение вероятности защищенности VoIP сети путем уменьшения времени, необходимого для проведения анализа динамики действий нарушителя, анализа и обработки рисков.

Задачей исследования является разработка метода управления ИБ для VoIP сети с прогнозированием воздействий на основе интеллектуальных технологий [5]. Для этого необходимо декомпозировать задачу на частные подзадачи: разработать структуру системы управления (СУ) ИБ, интегрируемой в VoIP сеть; разработать структурные схемы модулей распознавания вторжения с прогнозированием и оценки рисков ИБ за счет внедрения интеллектуальной системы; разработать алгоритм метода управления ИБ VoIP сети; разработать модели атак на сеть VoIP-телефонии.

Метод обеспечения информационной безопасности сети VoIP-телефонии

Распределенная VoIP сеть (рис. 1) может подвергаться информационному вторжению со стороны нарушителя.

Предлагаемый метод обеспечения ИБ содержит взаимоувязанную последовательность процессов [3]. Предложено обнаруживать, анализировать действия и прогнозировать реализацию атак на VoIP сеть с последующим блокированием нарушителя.

Аналогично [6] в целях сокращения времени реагирования СЗИ ($t_{СЗИ}$) предложена структурная схема СУ ИБ (рис. 2) с учетом внедрения интеллектуальной системы обнаружения вторжений (ИСОВ) [7]. Данные об инцидентах безопасности собираются с оборудования сети и СЗИ с помощью протоколов обмена служебной информацией. После чего проводятся унификация, фильтрация и установление приоритета полученной информации о безопасности сети. Далее оценивается защищенность сети на основе данной структурированной информации. Одновременно с этим строится прогноз воздействия и риска от него [8–10].

Нижний уровень работы ИСОВ предполагает обработку событий, происходящих в сегменте VoIP сети, и принятие мер в соответствии с их влиянием на ИБ с использованием модуля, представленного на рис. 3, *а* [11]. Данный модуль позволяет провести обработку возникающих или изменяющихся рисков.

Верхний уровень ИСОВ предполагает прогнозирование воздействия на VoIP сети с использованием другого модуля (рис. 3, *б*).

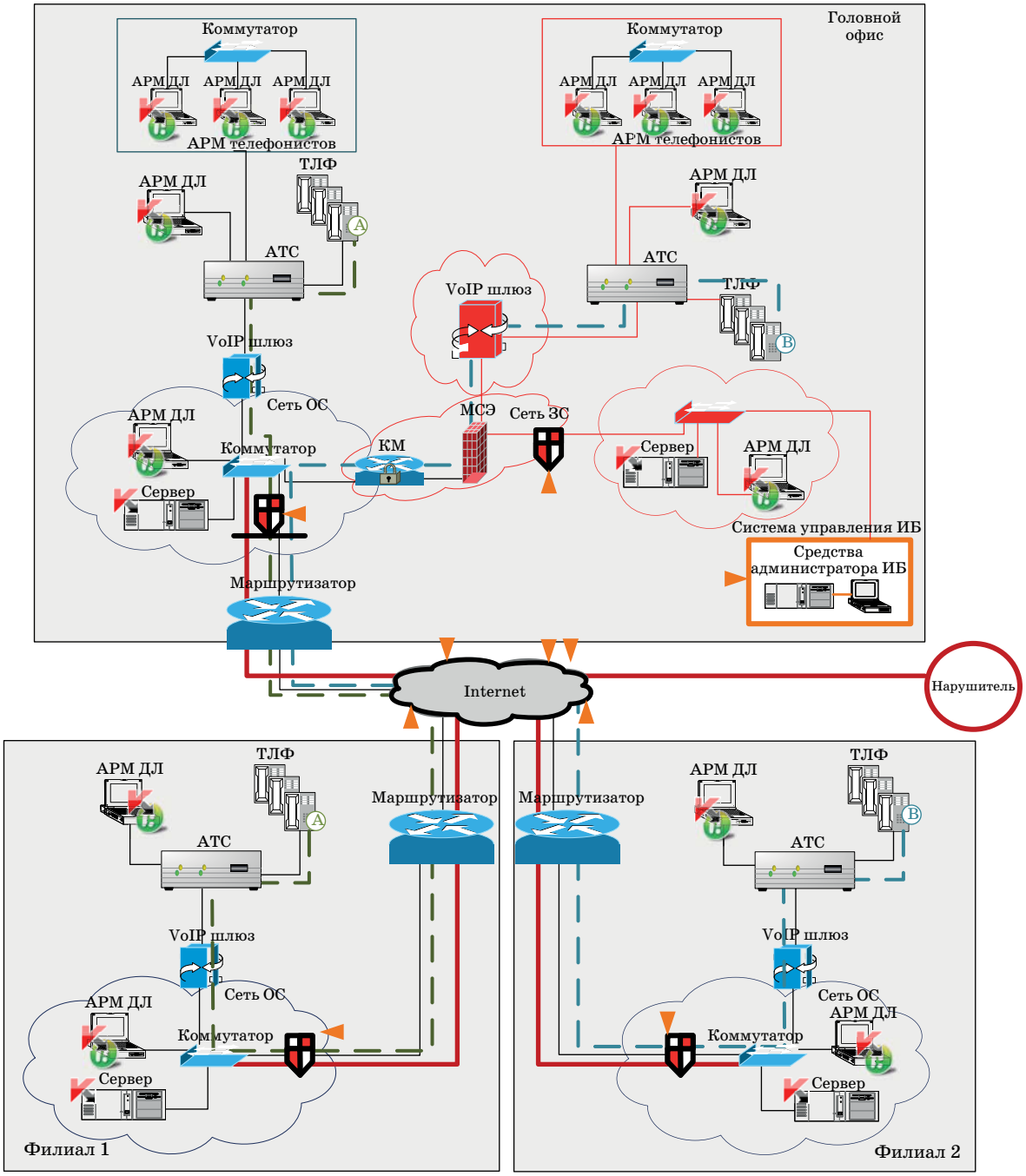
Прогнозирование воздействия и риска от него предлагается реализовать на основе интеллектуальных технологий, а именно модулярной гибридной системы прогнозирования временных рядов. Суть заключается в том, что прогнозы сравниваются по критериям, например точности, актуальности или соответствию горизонту прогнозирования [12].

В целях выявления уязвимостей VoIP сети на основе тестирования сети и пополнения базы данных (БД) угрозами несанкционированного действия (НСД) за счет анализа действий противника в ложной инфраструктуре VoIP сети разработан алгоритм метода обеспечения ИБ VoIP сети (рис. 4). Он позволяет реализовать наблюдение, выделение признаков атаки в пакетах сообщений, поступающих в VoIP сеть, и распознавание вторжения с выбором реализации способа защиты.

Системе управления ИБ необходимо произвести сбор информации о уязвимостях VoIP сети, текущей защищенности, сбор статистики и анализ действий нелегитимного пользователя, прогнозирование уровня защищенности с учетом анализа динамики действий нелегитимного пользователя, выработку стратегий противодействия и рекомендаций по корректировке работы СЗИ.

Уязвимость VoIP сети — это свойства VoIP сети и СЗИ, при которых нелегитимный пользователь методом вторжения нарушит защищенность информации. Поэтому в целях эффективного управления ИБ и своевременного устранения угроз безопасности предложено выявлять уязвимости на этапе, когда злоумышленник находится вне реальной сети, запущенной в VoIP сети в процессе ее функционирования.

Суть данного решения заключается в том, что идет получение пакетов сообщений от нелегитимных пользователей, которые не идентифицируются как угроза ИБ, а также в целях пополнения БД известных угроз в СУ ИБ формируются массивы, имитирующие ресурсы VoIP сети [13]. Также формируются виртуальные контейнеры на выделенном сервере в СУ ИБ на основе данных о VoIP сети и запускаются в работу в режиме функционирования VoIP сети. Принимаются на выделенном сервере пакеты сообщений от нелегитимного пользователя, пока соединение с нелегитимным пользователем не будет разорвано. Нелегитимный пользователь, работая с данным сервером, предполагает, что он находится в реальной VoIP сети, и проводит подготовку и реализацию вторжения. Действия нелегитимного пользователя ре-

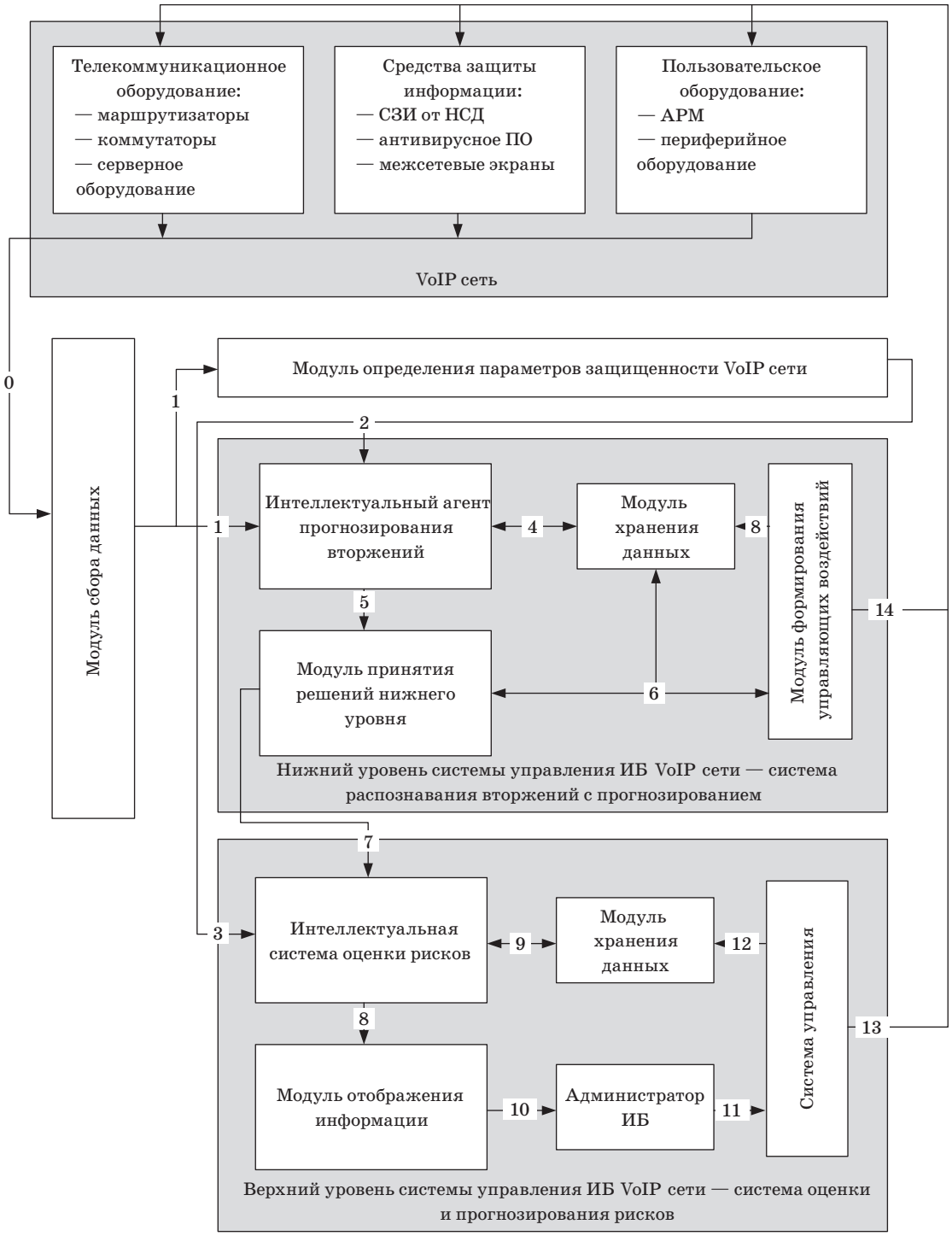


	Процессы реализации управления ИБ верхнего уровня
	Маршруты следования IP-трафика
	Воздействие нарушителя
Сеть ОС	Сеть открытого сегмента
Сеть ЗС	Сеть закрытого сегмента
АТС	Автоматическая телефонная станция
ТЛФ	Телефонный аппарат
АРМ ДЛ	Автоматизированное рабочее место должностного лица

Система управления ИБ	ОС	ЗС
Оборудование управления ИБ транспортной сети (нижнего уровня)	+	+
Средства администратора системы управления ИБ (верхнего уровня)	+	+

Стандартные СЗИ	ОС	ЗС
Антивирусное программное обеспечение	+	+
Средства разграничения доступа	+	+
Криптомаршрутизатор (КМ)	–	+
Межсетевой экран (МСЭ)	–	+

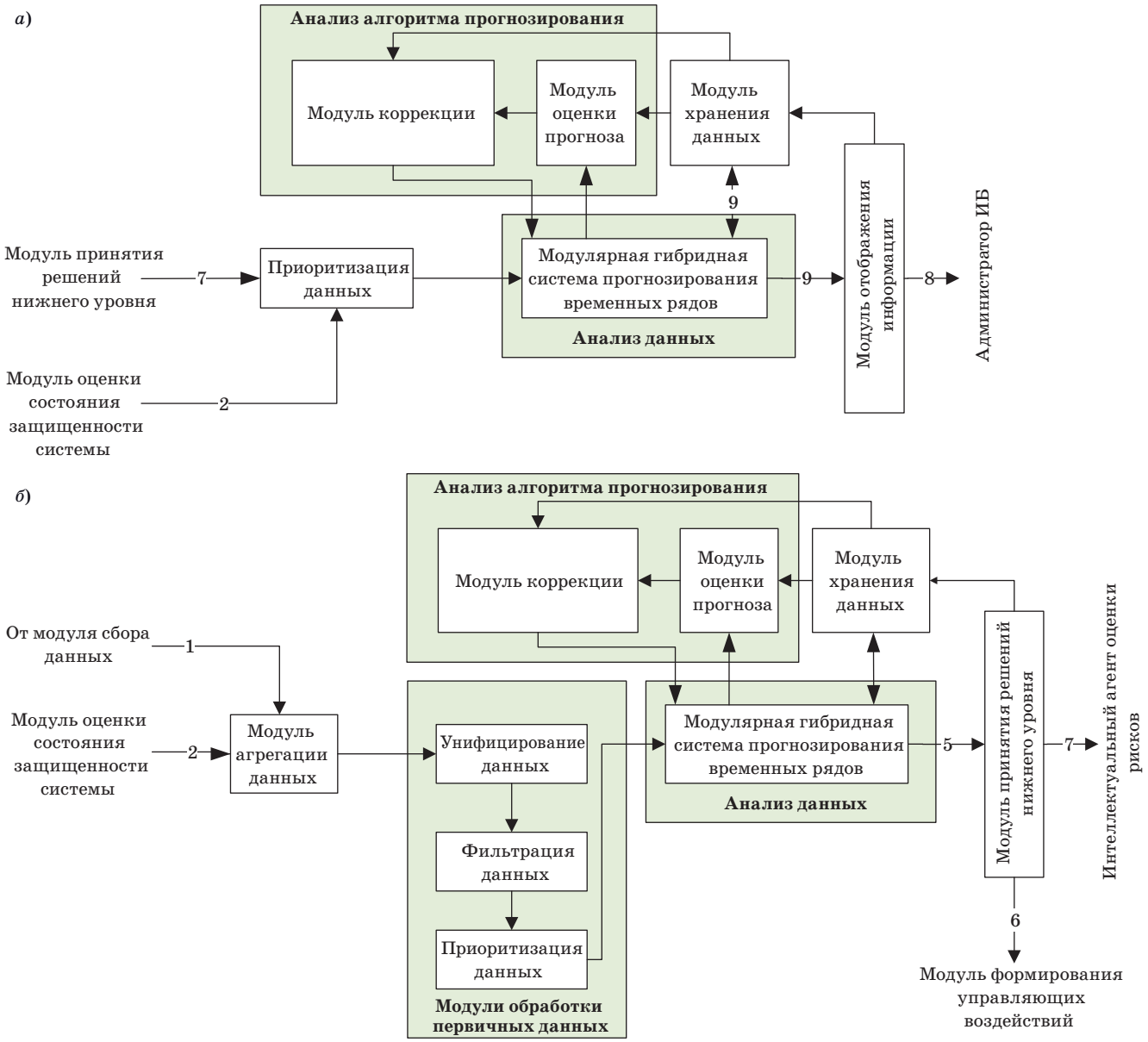
■ ***Рис. 1.*** VoIP сеть с интегрированной в нее СУ ИБ
■ ***Fig. 1.*** VoIP network with integrated IS management system



■ ***Рис. 2.*** Система управления ИБ с использованием ИСОВ
■ ***Fig. 2.*** Information security management system using intelligent intrusion detection system

гистрируются. В случае реализации угроз ИБ данные по действиям нелегитимного пользователя регистрируются и записываются в массив как новая угроза ИБ, тем самым пополняя БД. Далее проводится анализ, на основе которого принимается решение по изменению настроек СЗИ сети VoIP.

Анализ действий противника на сети VoIP осуществляется за счет моделирования атак на основе теории графов.



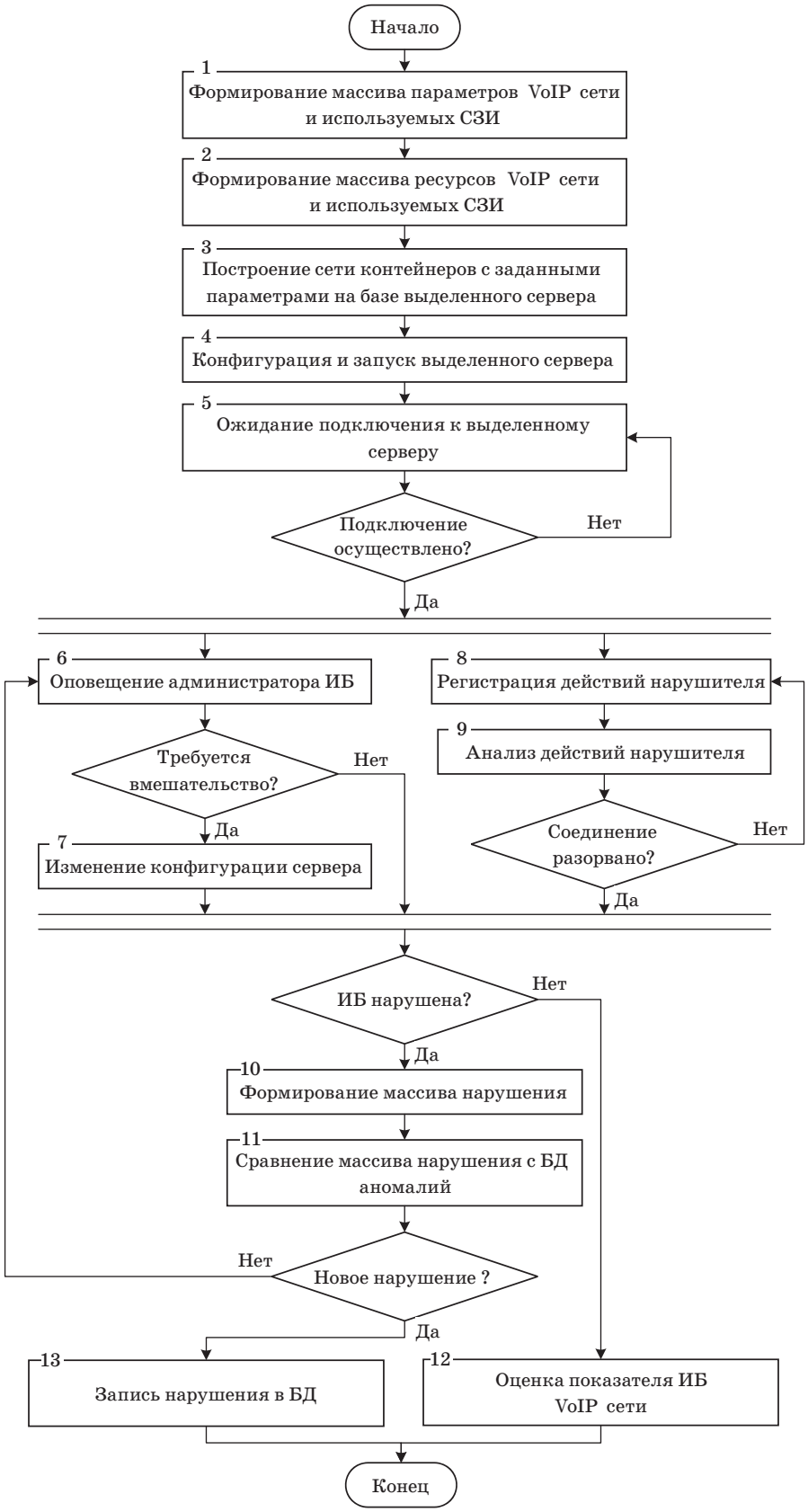
■ ***Рис. 3.*** Структура модуля интеллектуальной оценки рисков (*а*) и прогнозирования воздействия (*б*)
■ ***Fig. 3.*** The structure of the intelligent risk assessment module (*а*) and impact prediction module (*б*)

Модели действий нарушителя

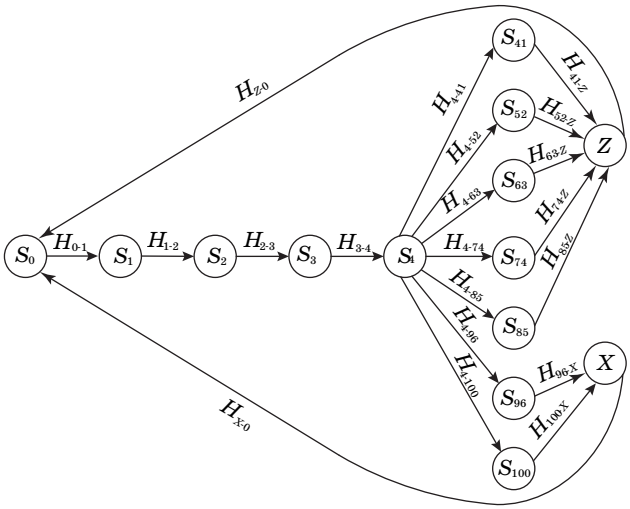
Одной из задач обеспечения ИБ VoIP сети является анализ воздействия, результатом которого являются параметры атаки. Данное действие необходимо для оценки эффективности системы обеспечения ИБ. В результате строится граф реализации угроз [14] и формируются рекомендации по корректировке настроек СЗИ.

Учтены нарушители в виде обычного посетителя организации и администратора информационной сети, который подключает ЭВМ. Если нарушитель первого типа знает логин и пароль от администрирования сети, то сразу авторизуется на сервере. Если не знает, то он подбирает логин и пароль (брутфорс (англ. *brute force*) — атака путем автоматического перебора паролей (иногда связки логинов-паролей), наиболее длительный и простой вид взлома) и авторизуется на сервере. После чего нарушитель администрирует свое устройство в сегменте сети и начинает сканировать сеть SIP (Session Initiation Protocol) сканером и находит список VoIP-устройств. После удачного поиска VoIP-устройств реализует атаку. На рис. 5 представлен граф реализации угроз VoIP сети [15].

Состояния нарушителя при реализации угроз описаны в табл. 1.



■ ***Рис. 4.*** Алгоритм метода обеспечения ИБ VoIP сети
■ ***Fig. 4.*** Algorithm of the VoIP network security method



■ ***Рис. 5***. Граф реализации угроз
■ ***Fig. 5***. Threat realization graph

■ ***Таблица 1***. Состояния нарушителя при реализации угроз
■ ***Table 1***. The state of the intruder during the implementation of threats

Состояние	Описание
S_0	Исходное состояние
S_1	Подготовка к воздействию
S_2	Сканирование нарушителем ИВС на наличие VoIP-устройств с помощью SIP-сканера
S_3	Сбор информации об объекте воздействия
S_4	Выбор осуществления типа атаки
S_{41}	Засорение канала передачи данных путем спама
S_{52}	Нагрузка с большим воздействием на VoIP-оборудование
S_{63}	Отправка постоянных пакетов данных на сеть VoIP
S_{74}	Создание виртуальных VoIP-устройств
S_{85}	Изменение ID-устройства
S_{96}	Перехват голосовых и медианных пользователей
S_{100}	Изменение пароля администратора ИВС
Z	Реализация отказа в обслуживании
X	Реализация угрозы хищения информации

Рассмотрены две атаки: человек посередине (Man in the Middle, MiTM) и спам для интернет-телефонии (Spam Over Internet Telephony, SPIT), так как они являются часто используемыми и наносящими большой материальный ущерб предприятию [16].

1. *Модель процесса нарушения безопасной передачи информации в сетях VoIP-телефонии при атаке MiTM*.

Перехват данных — самая большая проблема в сетях VoIP-телефонии, обусловлено это тем, что данные в стеке протоколов TCP/IP передаются в открытом виде [17]. Нарушителем выделяется маршрутизатор, который образует беспроводную сеть Wi-Fi. Определяется машина, на которой установлен анализатор трафика, например WireShark. Далее через netsh подготавливается точка доступа SSID (Service Set Identifier). Нарушитель в ходе разведки получает учетные данные для создания общей точки доступа с таким же названием и таким же паролем доступа, как на машине с WireShark. После чего перезагружается роутер, чтобы отключить всех клиентов. В этот момент включается двойник доступа. В итоге сетевое взаимодействие осуществляется через двойника доступа. Далее запускается WireShark и прослушивается трафик от клиентов.

Состояния нарушителя при реализации угрозы MiTM описаны в табл. 2.

■ ***Таблица 2***. Состояния нарушителя при реализации угрозы MiTM
■ ***Table 2***. The state of the intruder when the MiTM threat is implemented

Состояние	Описание
S_0	Исходное состояние
S_1	Постановка задачи на воздействие
S_2	Генерация пароля для входа в систему
S_3	Вход в систему под учетной записью «администратор»
S_4	Просмотр пароля и имени точки доступа на роутере для дальнейшего создания двойника на своем устройстве
S_5	Имитация ложной точки доступа
S_6	Обращение в Интернет через ложную точку доступа
S_7	Вывод из строя роутера с помощью DDoS-атаки
S_8	Ввод в сеть ложной точки доступа вместо атакуемого роутера
X	Реализация угрозы хищения информации

2. *Модель процесса нарушения безопасной передачи информации в сетях VoIP-телефонии при атаке SPIT.*

Данный вид воздействия возможен при помощи программ дозвона, таких как Spitter, по номерам из БД, с донесением заранее записанного голосового сообщения, с заранее пройденной авторизацией с помощью похищенных логинов и паролей от SIP учетных записей [18]. В отличие от спама, рассылаемого по e-mail в виде сообщений, вероятность прослушивания голосовых сообщений пользователями считается выше.

Состояния нарушителя при реализации угрозы SPIT описаны в табл. 3.

Разработана математическая модель активного нарушителя (рис. 6) [15], позволяющая рассчитать вероятностно-временные характеристики атаки в зависимости от значений вероятностей промежуточных атак [19, 20].

Определены вероятности нахождения нарушителя в различных состояниях реализации угрозы P_0, ..., P_8:

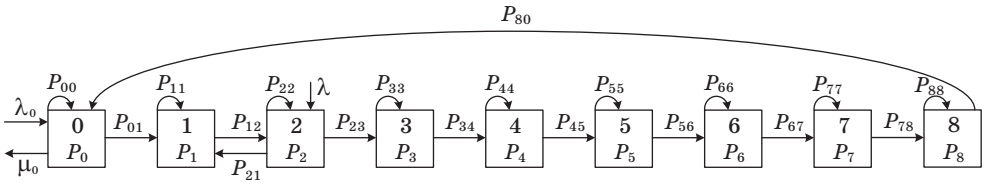
$$\begin{cases} P_0 = \dfrac{1}{1+\alpha\dfrac{\mu_0}{\beta_7}\left(1+\dfrac{\beta_7}{\beta_6}\left(1+\dfrac{\beta_6}{\beta_5}\left(1+\dfrac{\beta_5}{\beta_4}\left(1+\dfrac{\beta_4}{\beta_3}\left(1+\dfrac{\beta_3}{\beta_2}\left(1+\dfrac{\lambda+\beta_2}{\beta_1}\right)\right)\right)\right)\right)\right)}; \\ P_1 = \left(\alpha\cdot\dfrac{(\lambda+\beta_2)\beta_3\beta_4\beta_5\beta_6\beta_7\mu_0}{\beta_1\beta_2\beta_3\beta_4\beta_5\beta_6\beta_7}\right)\cdot\left(\dfrac{1}{1+\alpha\dfrac{\mu_0}{\beta_7}\left(1+\dfrac{\beta_7}{\beta_6}\left(1+\dfrac{\beta_6}{\beta_5}\left(1+\dfrac{\beta_5}{\beta_4}\left(1+\dfrac{\beta_4}{\beta_3}\left(1+\dfrac{\beta_3}{\beta_2}\left(1+\dfrac{\lambda+\beta_2}{\beta_1}\right)\right)\right)\right)\right)\right)}\right); \\ \ldots \\ P_8 = \alpha\cdot\left(\dfrac{1}{1+\alpha\dfrac{\mu_0}{\beta_7}\left(1+\dfrac{\beta_7}{\beta_6}\left(1+\dfrac{\beta_6}{\beta_5}\left(1+\dfrac{\beta_5}{\beta_4}\left(1+\dfrac{\beta_4}{\beta_3}\left(1+\dfrac{\beta_3}{\beta_2}\left(1+\dfrac{\lambda+\beta_2}{\beta_1}\right)\right)\right)\right)\right)\right)}\right), \end{cases} \quad (1)$$

где $\alpha = \dfrac{\lambda_0}{\mu_0}$, λ_0 — плотность потока задач на вторжение; μ_0 — плотность потока успешных вторжений; β_1, ..., β_7 — плотность выполнения задач нарушителем по вторжению согласно состояниям при MiTM- и SPIT-атаках.

■ ***Таблица 3.*** Состояния нарушителя при реализации угрозы SPIT

■ ***Table 3.*** The state of the intruder when the SPIT threat is implemented

Состояние	Описание
S_0	Исходное состояние
S_1	Постановка задачи на воздействие
S_2	Генерация пароля для входа в систему
S_3	Вход в систему под учетной записью «администратор»
S_4	Поиск SIP-абонентов
S_5	Обнаружение SIP-телефонов в сети и в режиме офлайн
S_6	Хищение данных для авторизации учетных SIP-записей
S_7	Вход в систему с похищенных учетных SIP-записей
S_8	Запуск программного обеспечения (ПО) Spitter
H	Реализация угрозы хищения информации



■ ***Рис. 6***. Граф состояния нарушителя при реализации угроз MiTM и SPIT на VoIP сеть
■ ***Fig. 6***. The intruder's state graph when the MiTM and SPIT threat is implemented on the VoIP network

Вероятности воздействия $P_в = P_0 + P_8$, так как, с одной стороны, нарушитель в «состоянии 8» успешно завершает реализацию угроз MiTM и SPIT на VoIP сеть, после чего, вероятнее всего, принимает решение на следующее воздействие на защищаемую инфраструктуру, с другой стороны, без корректной работы нарушителя в «состоянии 0» реализация угроз MiTM и SPIT на VoIP сеть невозможна [13]. С учетом вышесказанного система (1) примет вид

$$P_в = \frac{1+\alpha}{1+\alpha\frac{\mu_0}{\beta_7}\left(1+\frac{\beta_7}{\beta_6}\left(1+\frac{\beta_6}{\beta_5}\left(1+\frac{\beta_5}{\beta_4}\left(1+\frac{\beta_4}{\beta_3}\left(1+\frac{\beta_3}{\beta_2}\left(1+\frac{\lambda+\beta_2}{\beta_1}\right)\right)\right)\right)\right)\right)}. \quad (2)$$

Получив значение вероятности воздействия, можно вычислить показатель (вероятность) защищенности VoIP сети ($P_{защ}$):

$$P_{защ} = 1 - P_в, \quad (3)$$

где $P_в$ — вероятность воздействия со стороны нарушителя на VoIP сеть.

Реализация моделей в виде программного обеспечения

Для оптимизации расчета воздействия на VoIP сети по выражению (2) и принятия оперативных мер по устранению нарушения целостности сети было принято решение разработать ПО, позволяющее упростить все расчеты и получить готовый результат.

Программное обеспечение было написано на языке Java ver.8. с применением платформы JavaFX (рис. 7).

С помощью ПО [21] на основе исходных данных из табл. 4, являющихся экспертными оценками, были проведены контрольные решения оценки вероятности защищенности VoIP сети в условиях реализации подбора пароля с различной интенсивностью от 3 до 9 мин (рис. 8, *а*), авторизации в системе администрирования роутера с интенсивностью от 0,2 до 0,6 мин (рис. 8, *б*) при MiTM, а также хищения данных для авторизации учетных SIP-записей с интенсивностью от 1 до 30 мин (рис. 8, *в*) и массового входа в систему с похищенных учетных SIP-записей с интенсивностью от 2 до 6 мин (рис. 8, *г*) при SPIT.

Анализ зависимостей на рис. 8, *а–г* позволяет сделать вывод о том, что увеличение времени реагирования СЗИ $t_{СЗИ}$ VoIP сети при атаках MiTM и SPIT понижает защищенность $P_{защ}$ ниже требуемого уровня 0,9. В связи с этим возможно определить требования к $t_{СЗИ}$ с учетом условий противоборства:

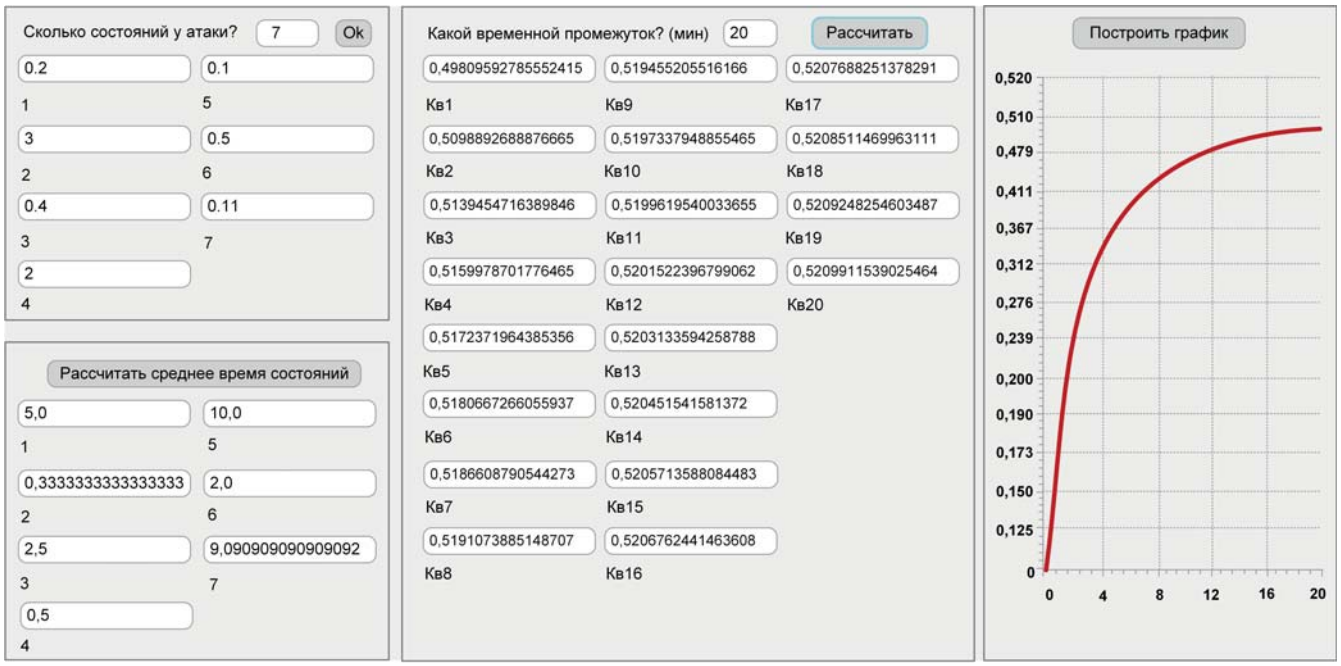
1) $t_{СЗИ} \le 0{,}0012$ мин в условиях реализации подбора пароля при MiTM (см. рис. 8, *а*);

2) $t_{СЗИ} \le 0{,}0014$ мин в условиях реализации авторизации в системе администрирования роутера при MiTM (см. рис. 8, *б*);

3) $t_{СЗИ} \le 0{,}046$ мин в условиях реализации получения из БД логинов и паролей от SIP учетных записей при SPIT (см. рис. 8, *в*);

4) $t_{СЗИ} \le 0{,}045$ мин в условиях реализации массовой авторизации учетных SIP-записей при SPIT (см. рис. 8, *г*).

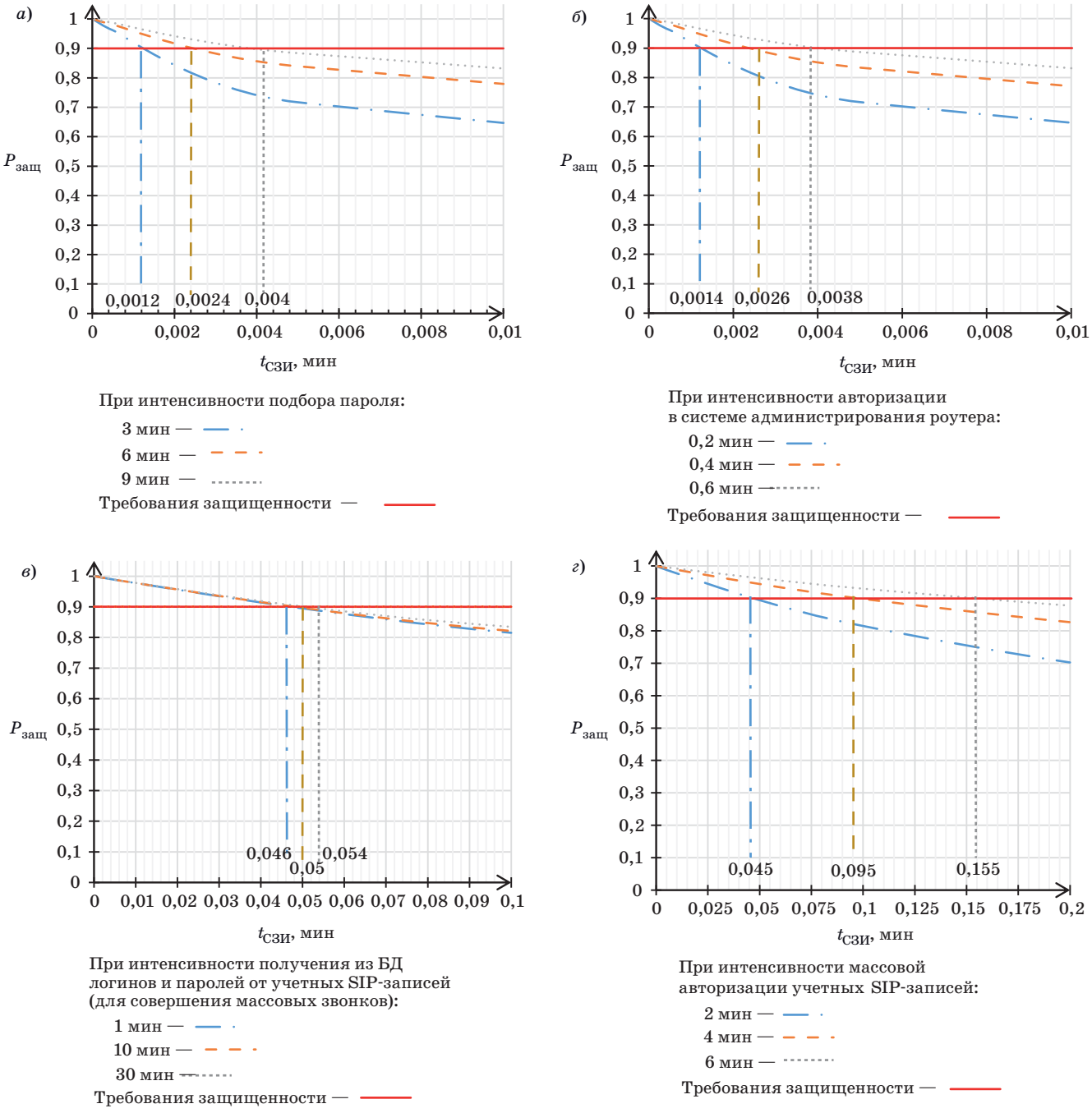
Также из графиков видно, что наибольшую угрозу для VoIP сети представляет MiTM-атака, так как вероятность защищенности резко падает в короткий промежуток времени. Следовательно, событие «реализация генерации пароля для входа в систему» будет более вероятным, в связи с этим вероятность того, что нарушитель этим воспользуется, высока. Поэтому администратору безопасности необходимо обратить внимание на устранение именно этой уязвимости [14].



■ ***Рис. 7.*** Интерфейс ПО для оперативного расчета вероятности воздействия на VoIP сети
■ ***Fig. 7.*** Software interface for on-line calculation of probability of impact on VoIP networks

■ ***Таблица 4.*** Исходные данные для контрольной оценки вероятности защищенности VoIP сети в условиях реализации MiTM и SPIT
■ ***Table 4.*** Initial data for the control assessment of the probability of VoIP network security in the context of MiTM and SPIT implementation

Событие	MiTM			SPIT		
	Описание	Параметр	Значение, мин	Описание	Параметр	Значение, мин
S_2	Генерация пароля (ГП) для входа в систему	$\beta_1 = \frac{1}{t_{ГП}}$	3 6 9	Генерация пароля для входа в систему	$\beta_1 = \frac{1}{t_{ГП}}$	10
S_3	Вход в систему (ВС) под учетной записью «администратор»	$\beta_2 = \frac{1}{t_{ВС}}$	0,2 0,4 0,6	Вход в систему под учетной записью «администратор»	$\beta_2 = \frac{1}{t_{ВС}}$	2
S_4	Просмотр пароля (ПП) и имени точки доступа на роутере для дальнейшего создания двойника на своем устройстве	$\beta_3 = \frac{1}{t_{ПП}}$	0,5	Поиск SIP-абонентов	$\beta_3 = \frac{1}{t_{SIP}}$	2
S_5	Имитация ложной точки доступа (ЛТД)	$\beta_4 = \frac{1}{t_{ЛТД}}$	0,6	Обнаружение (О) SIP-телефонов в сети и в режиме офлайн	$\beta_4 = \frac{1}{t_{О}}$	1
S_6	Обращение в Интернет (ОИ) через ложную точку доступа	$\beta_5 = \frac{1}{t_{ОИ}}$	0,3	Хищение данных (ХД) для авторизации учетных SIP-записей	$\beta_5 = \frac{1}{t_{ХД}}$	1 10 30
S_7	Вывод из строя роутера с помощью DDoS-атаки	$\beta_6 = \frac{1}{t_{DDoS}}$	1	Авторизация (А) в системе с похищенных учетных SIP-записей	$\beta_6 = \frac{1}{t_{А}}$	2 4 6
S_8	Подмена (П) атакуемого роутера ложной точкой доступа	$\beta_7 = \frac{1}{t_{П}}$	0,2	Запуск ПО Spitter	$\beta_7 = \frac{1}{t_{Spitter}}$	0,5



■ ***Рис. 8.*** Зависимость показателя защищенности VoIP сети от времени реагирования СЗИ в условиях: *а* — реализации подбора пароля при MiTM; *б* — реализации авторизации в системе администрирования роутера при MiTM; *в* — хищения данных для авторизации учетных SIP-записей при SPIT; *г* — реализации массового входа в систему с похищенных учетных SIP-записей при SPIT

■ ***Fig. 8.*** Dependence of the VoIP network security indicator on the response time of information security tools in conditions of: *а* — password matching in MiTM; *б* — authorization implementation in the router administration system with MiTM; *в* — data theft for authorization of SIP records during SPIT; *г* — a mass logon from stolen SIP accounts at SPIT

Заключение

Представлен анализ VoIP сети организации, алгоритм контроля ситуационных параметров при стохастической неопределенности, предложена архитектура прототипа VoIP сети, а также ПО, позволяющее оптимизировать расчеты для быстрого получения результатов оценки защищенности VoIP сети в целях последующего принятия решений по защите целостности сети.

Данный метод включает обнаружение, анализ действия нарушителя и прогноз реализации атак

на VoIP сеть с последующим блокированием нежелательной активности.

Научная новизна заключается в том, что в отличие от известных метод обеспечения ИБ сети VoIP-телефонии учитывает: выявление уязвимостей VoIP сети на основе тестирования реальной сети; выявление уязвимостей и пополнение БД угрозами НСД за счет анализа действий противника в ложной инфраструктуре VoIP сети; профили и виды атак, направленных на VoIP сеть и описанных в виде графов.

Теоретическая значимость заключается в разработке и усовершенствовании известных моделей, что позволяет определять вероятность воздействия, структуру и профиль атаки, а также способы защиты VoIP сети.

Практическая значимость определяется возможностью использовать метод управления обеспечения ИБ VoIP сети при разработке СУ ИБ.

Получены результаты аналитического моделирования, которые показали, что предложенный подход обеспечивает требуемый уровень достоверности принимаемых решений, что позволяет повысить вероятностно-временные характеристики работы VoIP сети.

В дальнейшем будут исследоваться возможности по реализации данного метода в автоматизированном виде с применением нейронной сети, позволяющей полностью на программном уровне принимать самостоятельные решения по предотвращению хищения информации, нарушения целостности и доступности сети.

Литература

1. **Маликов А. В., Авраменко В. С., Саенко И. Б.** Модель и метод диагностирования компьютерных инцидентов в информационно-коммуникационных системах, основанные на глубоком машинном обучении. *Информационно-управляющие системы*, 2019, № 6, с. 32–42. doi:10.31799/1684-8853-2019-6-32-42
2. **Котлякова В. В., Кузьмина И. В.** Автоматизация процесса функционального тестирования распределенной информационной системы с использованием дистрибутива Docker. *Информационные системы и технологии ИСТ-2020: сб. матер. XXVI Междунар. науч.-техн. конф.*, Н. Новгород, 24–28 апреля 2020 г. Н. Новгород, 2020, с. 1247–1250.
3. **Котенко И. В., Хмыров С. С.** Анализ актуальных методик атрибуции нарушителей кибербезопасности при реализации целевых атак на объекты критической инфраструктуры. *Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2021): сб. науч. ст. X Юбилейной Междунар. науч.-техн. и науч.-метод. конф.*, Санкт-Петербург, 24–25 февраля 2021 г. СПб., 2021, т. 4, с. 536–541.
4. Пат. 2705773 РФ, C1 G 06 F 12/14. *Способ защиты информационно-вычислительной сети от вторжений*, В. А. Липатников (РФ), К. В. Чепелев (РФ), А. А. Шевченко (РФ). № 2019100252; заявл. 09.01.2019; опубл. 11.11.2019, Бюл. № 32, 19 с.
5. **Липатников В. А., Тихонов В. А.** Распознавание вторжений нарушителя при управлении кибербезопасностью инфраструктуры интегрированной организации на основе нейро-нечетких сетей и когнитивного моделирования. *Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019): сб. науч. ст. VIII Междунар. науч.-техн. и науч.-метод. конф.*, Санкт-Петербург, 27–28 февраля 2019 г. СПб., 2019, т. 4, с. 659–664.
6. **Котенко И. В., Ушаков И. А., Пелевин Д. В., Преображенский А. И., Овраменко А. Ю.** Выявление инсайдеров в корпоративной сети: подход на базе UBA и UEBA. *Защита информации. Инсайд*, 2019, № 5 (89), с. 26–35.
7. **Костарев С. В., Карганов В. В., Липатников В. А.** *Технологии защиты информации в условиях кибернетического противоборства*. Санкт-Петербург, ВАС им. Буденного, 2020. 716 с.
8. **Chandra K. P. B., Gu D.** *Nonlinear Filtering: Methods and Applications*. Springer, 2019. 197 p.
9. **Федорченко А. В., Дойникова Е. В., Котенко И. В.** Автоматизированное определение активов и оценка их критичности для анализа защищенности информационных систем. *Тр. СПИИРАН*, 2019, т. 18, № 5, с. 1182–1211.
10. **Ярушева С. А., Аверкина А. Н., Федотова А. В.** Модулярная модель прогнозирования временных рядов на основе нейро-нечетких сетей и когнитивного моделирования. *Нечеткие системы и мягкие вычисления*, 2017, т. 12, № 2, с. 159–168. doi:10.26456/fssc31
11. **Guay M., Adetola V., DeHaan D.** *Robust and Adaptive Model Predictive Control of Nonlinear Systems*. The Institution of Engineering and Technology, 2016. 253 p. doi:10.1049/PBCE083E
12. **Rezaee Z., Dorestani A., Aliabadi S.** Application of time series analyses in big data: Practical, research, and education implications. *Journal of Emerging Technologies in Accounting*, 2018, vol. 15, iss. 1, pp. 183–197.
13. **Липатников В. А., Колмыков Д. В., Косолапов В. С.** Способ управления информационной безопасностью информационно-вычислительной сети при вторжениях типа распределенного отказа в обслуживании. *Состояние и перспективы развития современной науки по направлению «Информационная безопасность»: сб. ст. III Всерос. науч.-техн. конф.*, Анапа, 2021, с. 643–654.
14. **Сорокин М. А., Стародубцев Ю. И.** Методика обоснования количества и мест размещения средств сетевого контроля информационного обмена меж-

ду элементами корпоративной системы управления. *Вопросы оборонной техники. Сер. 16. Технические средства противодействия терроризму*, 2021, № 3-4 (153-154), с. 65–74.
15. **Липатников В. А., Сокол Д. С.** Модель нарушителя безопасной передачи информации в сетях VoIP-телефонии. *Транспорт России: проблемы и перспективы — 2020: матер. Юбилейной междунар. науч.-практ. конф.*, Санкт-Петербург, 10–11 ноября 2020 г. СПб., Институт проблем транспорта им. Н. С. Соломенко РАН, 2020, с. 187–192.
16. **Pallaprolu S. C., Sankineni R., Thevar M., Karabatis G., Wang J.** Zero-day attack identification in streaming data using semantics and spark. *Proc. of the IEEE Intern. Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, 2017, 25–30 June, pp. 121–128.
17. **Peters M. D., Wieder B., Sutton S. G., Wakefield J.** Business intelligence systems use in performance capabilities: Implications for enhanced competitive advantage. *International Journal of Accounting Information Systems*, 2021, no. 21, pp. 1–17.
18. **Borthick A. F., Schneider G. P., Viscelli T. R.** Analyzing data for decision making: Integrating spreadsheet modeling and database querying. *Issues in Accounting Education*, 2017, vol. 32, iss. 1, pp. 59–66. doi:10.2308/iace-51385
19. **Xi X., Zhang T., Ye W., Wen Z., Zhang S., Du D., Gao Q.** An ensemble approach for detecting anomalous user behaviors. *International Journal of Software Engineering and Knowledge Engineering*, 2018, vol. 28, no. 11-12, pp. 1637–1656. doi:10.1142/S0218194018400211
20. **Kawamura H.** *Advanced Process Control*. https://blog.yokogawa.com/ru-advanced-solutions-blog/-ru-advanced-process-control (дата обращения: 16.10.2019).
21. **Сокол Д. С., Косолапов В. С., Липатников В. А., Парфиров В. А., Шевченко А. А.** *Расчет коэффициентов воздействия атак на программно-аппаратное оборудование*. Свидетельство о регистрации программы для ЭВМ 2021616926, 29.04.2021. Заявка № 2021612805 от 05.03.2021.

UDC 004; 621.398
doi:10.31799/1684-8853-2022-1-54-67

Method for ensuring information security of a VoIP telephony network with a forecast of an intruder's intrusion strategy

V. A. Lipatnikov[a], Dr. Sc., Tech., Professor, orcid.org/0000-0002-3736-4743, lipatnikovanl@mail.ru
A. A. Shevchenko[a], Research Fellow, orcid.org/0000-0001-9113-1089
V. S. Kosolapov[a], Post-Graduate Student, orcid.org/0000-0001-8464-779X
D. S. Sokol[a], Science Company Operator, orcid.org/0000-0002-1532-8872
[a]S. M. Budenny Military Academy of Communication, 3, Tikhoretskii Pr., 190064, Saint-Petersburg, Russian Federation

Introduction: The development of technologies in the field of information and telecommunications, as well as the unification of networks, and in particular the construction of distributed VoIP telephony networks, allow us to formulate the problem that the known methods of managing the protection of VoIP networks are not effective enough in modern conditions, since they take into account only one side of the information confrontation. **Purpose:** To develop a method for ensuring the information security of a VoIP telephony network, which allows to increase the probability of VoIP network security by reducing the time required for analyzing the actions of the violator, analyzing and processing risks under the influence of the violator. **Results:** Based on the proposed structure of an information security management system integrated into a VoIP network, a method for ensuring the information security of a VoIP telephony network under the influence of an intruder has been developed by introducing decision-making support processes in the VoIP network information security management system using intelligent intrusion detection tools distributed across segments. This method allows you to build a graph of events of the intruder's actions, on the basis of which mathematical modeling of MiTM and SPIT attacks on the VoIP telephony network is carried out. As a result of the simulation, the dependence of the successful impact on the internal and external characteristics of attacks is obtained, which is the main one of the developed software, which allows to obtain the values of the probability of security of the VoIP network from the parameters of the intruder's impact for further selection of adequate measures for managing the information security of the VoIP telephony network. The method includes the processes of analyzing the digital stream and determining the parameters of protocols and profiles of intruder attacks. **Practical relevance:** The developed method provides an opportunity to study issues aimed at the security of the VoIP-telephony network, which is affected by violators.

Keywords — VoIP networks, information security, MiTM, SPIT, Markov random process, attack model.

For citation: Lipatnikov V. A., Shevchenko A. A., Kosolapov V. S., Sokol D. S. Method for ensuring information security of a VoIP telephony network with a forecast of an intruder's intrusion strategy. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 54–67 (In Russian). doi:10.31799/1684-8853-2022-1-54-67

References

1. Malikov A. V., Avramenko V. S., Saenko I. B. Model and method for diagnosing computer incidents in information and communication systems based on deep machine learning. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 6, pp. 32–42 (In Russian). doi:10.31799/1684-8853-2019-6-32-42
2. Kotlyakova V. V., Kuzmina I. V. Automation of the functional testing process of a distributed information system using the docker distribution. *Materialy XXVI Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii "Informatsionnyye sistemy i tekhnologii" IST-2020* [Materials of the XXVI Intern. Sci. and Techn. Conf. "Information Systems and

Technologies" IST-2020]. N. Novgorod, 2020, pp. 1247–1250 (In Russian).
3. Kotenko I., Khmyrov S. Analysis of current methods of attributing cyber security offensers in the implementation of targeted attacks on objects of critical infrastructure. *10th Intern. Conf. on Advanced Infotelecommunications (ICAIT 2021)*, Saint-Petersburg, 2021, vol. 4, pp. 536–541 (In Russian).
4. Lipatnikov V. A., Chepelev K. V., Shevchenko A. A. *Sposob zashchity informacionno-vychislitel'noj seti ot vtorzhenij* [Method of protecting an information network from intrusions]. Patent RF, no. 2705773, 2019.
5. Lipatnikov V. A., Tikhonov V. A. Recognition of offenders actions in the management of cyber security of the integrated organization infrastructure on the basis of neuro-fuzy networks and cognitive modeling. *8th Intern. Conf. on Advanced Infotelecommunications (ICAIT 2019)*, Saint-Petersburg, 2019, vol. 4, pp. 659–664 (In Russian).
6. Kotenko I. V., Ushakov I. A., Pelevin D. V., Preobrazhenskiy A. I., Ovramenko A. U. Identification of insiders in the corporate network: UBA and UEBA based approach. *Zaśita Informacii. Inside*, 2019, no. 5 (89), pp. 26–35 (In Russian).
7. Kostarev S. V., Karganov V. V., Lipatnikov V. A. *Tekhnologii zashchity informatsii v usloviyakh kiberneticheskogo protivoborstva* [Technologies of information protection in the conditions of cybernetic confrontation]. Saint-Petersburg, VAS im. Budennogo Publ., 2020. 716 p. (In Russian).
8. Chandra K. P. B., Gu D. *Nonlinear Filtering: Methods and Applications*. Springer, 2019. 197 p.
9. Fedorchenko A. V., Doynikova E. V., Kotenko I. V. Automated detection of assets and calculation of their criticality for the analysis of information system security. *SPIIRAS Proc.*, 2019, vol. 18, no. 5, pp. 1182–1211 (In Russian). doi:10.15622/sp.2019.18.5.1182-1211
10. Yarusheva S. A., Averkina A. N., Fedotova A. V. Modular model for time series forecasting based on neuro-fuzzy nets and cognitive modelling. *Nechetkie sistemy i myagkie vychisleniya*, 2017, vol. 12, no. 2, pp. 159–168 (In Russian). doi:10.26456/fssc31
11. Guay M., Adetola V., DeHaan D. *Robust and Adaptive Model Predictive Control of Nonlinear Systems*. The Institution of Engineering and Technology, 2016. 253 p. doi:10.1049/PBCE083E
12. Rezaee Z., Dorestani A., Aliabadi S. Application of time series analyses in big data: Practical, research, and education implications. *Journal of Emerging Technologies in Accounting*, 2018, vol. 15, iss. 1, pp. 183–197.
13. Lipatnikov V. A., Kolmykov D. V., Kosolapov V. S. Sposob upravleniya informatsionnoy bezopasnost'yu informatsionno-vychislitel'noy seti pri vtorzheniyakh tipa raspredelennogo otkaza v obsluzhivanii. *Sbornik statey III Vserossiyskoy nauchno-tekhnicheskoy konferentsii "Sostoyaniye i perspektivy razvitiya sovremennoy nauki po napravleniyu "Informatsionnaya bezopasnost'"* [Collection of articles of the III All-Russian Scient. and Techn. Conf. "The state and prospects for the development of modern science in the direction of "Information security"]. Anapa, 2021, pp. 643–654 (In Russian).
14. Sorokin M. A., Starodubcev J. I. Methodology for substantiating the number and locations of network control facilities for information exchange between elements of the corporate management system. *Military Enginery. Scientific and Technical Journal. Counter-terrorism technical devices. Issue 16*, 2021, no. 3-4 (153-154), pp. 65–74 (In Russian).
15. Lipatnikov V. A., Sokol D. S. Intruder model for secure data transmission in VoIP telephony networks. *Mezhdunarodnaya nauchno-prakticheskaya konferentsiya "Transport Rossii: problemy i perspektivy — 2020"* [Intern. Scient. and Pract. Conf. "Transport of Russia: Problems and prospects — 2020"]. Saint-Petersburg, 2020, pp. 187–192 (In Russian).
16. Pallaprolu S. C., Sankineni R., Thevar M., Karabatis G., Wang J. Zero-day attack identification in streaming data using semantics and spark. *Proc. of the IEEE Intern. Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, 2017, 25–30 June, pp. 121–128.
17. Peters M. D., Wieder B., Sutton S. G., Wakefield J. Business intelligence systems use in performance capabilities: Implications for enhanced competitive advantage. *International Journal of Accounting Information Systems*, 2021, no. 21, pp. 1–17.
18. Borthick A. F., Schneider G. P., Viscelli T. R. Analyzing data for decision making: Integrating spreadsheet modeling and database querying. *Issues in Accounting Education*, 2017, vol. 32, iss. 1, pp. 59–66. doi:10.2308/iace-51385
19. Xi X., Zhang T., Ye W., Wen Z., Zhang S., Du D., Gao Q. An ensemble approach for detecting anomalous user behaviors. *International Journal of Software Engineering and Knowledge Engineering*, 2018, vol. 28, no. 11-12, pp. 1637–1656. doi:10.1142/S0218194018400211
20. *Advanced Process Control*. Available at: https://blog.yokogawa.com/ru-advanced-solutions-blog/-ru-advanced-process-control (accessed 24 August 2019).
21. Sokol D. S., Kosolapov V. S., Lipatnikov V. A., Parfirov V. A., Shevchenko A. A. *Raschet koeffitsiyentov vozdeystviya atak na programmno-apparatnoye oborudovaniye* [Calculation of attack impact coefficients on software and hardware equipment]. Certificate of registration of the computer program 2021616926, 2021.

СВЕДЕНИЯ ОБ АВТОРАХ

АБДУЛЛИН Азат Марселевич

Ассистент Института компьютерных наук и технологий Санкт-Петербургского политехнического университета Петра Великого, исследователь лаборатории верификации и анализа программ JetBrains Research.
В 2018 году окончил Санкт-Петербургский политехнический университет Петра Великого по специальности «Информатика и вычислительная техника».
Является автором трех научных публикаций.
Область научных интересов — верификация и анализ программ, автоматическая генерация тестов.
Эл. адрес: abdullin@kspt.icc.spbstu.ru

БРАНИЦКИЙ Александр Александрович

Старший научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского института информатики и автоматизации РАН.
В 2012 году окончил с отличием Санкт-Петербургский государственный университет по специальности «Математическое обеспечение и администрирование информационных систем».
В 2018 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 35 научных публикаций.
Область научных интересов — искусственный интеллект, компьютерные сети, информационная безопасность, функциональное программирование, теория формальных языков и компиляторов.
Эл. адрес: branitskiy@comsec.spb.ru

ИЦЫКСОН Владимир Михайлович

Директор Высшей школы интеллектуальных систем и суперкомпьютерных технологий Института компьютерных наук и технологий Санкт-Петербургского политехнического университета Петра Великого, руководитель лаборатории верификации и анализа программ JetBrains Research.
В 1996 году окончил Санкт-Петербургский государственный политехнический университет по специальности «Информатика и вычислительная техника».
В 2000 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 100 научных публикаций.
Область научных интересов — формальные методы, верификация и анализ программ, формальные спецификации, качество программного обеспечения.
Эл. адрес: vlad@icc.spbstu.ru

БАЛОНИН Николай Алексеевич

Профессор кафедры вычислительных систем и сетей Санкт-Петербургского государственного университета аэрокосмического приборостроения.
В 1982 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Автоматика и телемеханика».
В 2008 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 100 научных публикаций, в том числе трех монографий.
Область научных интересов — теория динамических систем, теория идентификации, теория операторов, теория матриц, вычислительные методы, интернет-робототехника, интернет-книги с исполняемыми алгоритмами, научные социальные сети.
Эл. адрес: korbendfs@mail.ru

ГРЫЗУНОВ Виталий Владимирович

Доцент кафедры информационных технологий и систем безопасности Российского государственного гидрометеорологического университета, Санкт-Петербург.
В 1999 году окончил Военную инженерно-космическую академию им. А. Ф. Можайского по специальности «Вычислительные машины, комплексы, системы и сети».
В 2004 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 80 научных публикаций.
Область научных интересов — адаптивное управление информационно-вычислительными системами в условиях дестабилизации, самоорганизующиеся информационно-вычислительные системы.
Эл. адрес: viv1313r@mail.ru

КОСОЛАПОВ Владислав Сергеевич

Адъюнкт Военной академии связи им. Маршала Советского Союза С. М. Буденного, Санкт-Петербург.
В 2011 году окончил Военную академию связи им. Маршала Советского Союза С. М. Буденного по специальности «Эксплуатация вычислительных машин, комплексов и сетей».
Является автором шести научных публикаций.
Область научных интересов — информационно-вычислительные системы, информационная безопасность.
Эл. адрес: kvs_mil@mail.ru

КОТЕНКО Игорь Витальевич

Профессор, главный научный сотрудник, заведующий лабораторией проблем компьютерной безопасности Санкт-Петербургского института информатики и автоматизации РАН.
В 1983 году окончил Военно-космическую академию им. А. Ф. Можайского по специальности «Математическое обеспечение автоматизированных систем управления», в 1987 году — Военную академию связи по специальности «Инженерная автоматизированных систем управления».
В 1999 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 500 научных публикаций.
Область научных интересов — безопасность компьютерных сетей, обнаружение компьютерных атак, межсетевые экраны и др.
Эл. адрес: ivkote@comsec.spb.ru

ЛИПАТНИКОВ Валерий Алексеевич

Профессор, старший научный сотрудник Военной академии связи им. Маршала Советского Союза С. М. Буденного, Санкт-Петербург, заслуженный изобретатель РФ, член-корреспондент РАЕН.
В 1974 году окончил Военную академию связи им. Маршала Советского Союза С. М. Буденного по специальности «Специальная радиотехника».
В 2000 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором 267 научных публикаций и 80 патентов на изобретения.
Область научных интересов — теория многоуровневой иерархической радиоэлектронной защиты, безопасности связи и информации инфотелекоммуникационных сетей.
Эл. адрес: lipatnikovanl@mail.ru

МОЛДОВЯН Дмитрий Николаевич

Научный сотрудник лаборатории кибербезопасности и постквантовых криптосистем Санкт-Петербургского института информатики и автоматизации РАН.
В 2009 году окончил Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» по специальности «Компьютерная безопасность».
В 2012 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором 79 научных публикаций и шести патентов на изобретения.
Область научных интересов — информационная безопасность, защита информации, криптосистемы с открытым ключом, постквантовая криптография, конечные некоммутативные алгебры.
Эл. адрес: mdn.spectr@mail.ru

КРАСОВ Андрей Владимирович

Доцент, заведующий кафедрой защищенных систем связи Санкт-Петербургского государственного университета им. проф. М. А. Бонч-Бруевича, почетный работник высшего профессионального образования.
В 1994 году окончил Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» по специальности «Автоматика и управление в технических системах».
В 2001 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 100 научных публикаций.
Область научных интересов — информационная безопасность, телекоммуникационные системы.
Эл. адрес: krasov@inbox.ru

МОЛДОВЯН Александр Андреевич

Профессор, главный научный сотрудник лаборатории кибербезопасности и постквантовых криптосистем Санкт-Петербургского института информатики и автоматизации РАН.
В 1974 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Автоматизированные системы управления».
В 2005 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 200 научных публикаций и 60 патентов на изобретения.
Область научных интересов — компьютерная безопасность, защита информации, криптография, протоколы электронной цифровой подписи.
Эл. адрес: maa1305@yandex.ru

МОЛДОВЯН Николай Андреевич

Профессор, заведующий научно-исследовательским отделом проблем информационной безопасности Санкт-Петербургского института информатики и автоматизации РАН, заслуженный изобретатель РФ.
В 1975 году окончил Кишиневский политехнический институт по специальности «Полупроводниковые приборы».
В 2001 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 250 научных публикаций и 60 патентов на изобретения.
Область научных интересов — информационная безопасность, криптография, электронная цифровая подпись, блочные шифры.
Эл. адрес: nmold@mail.ru

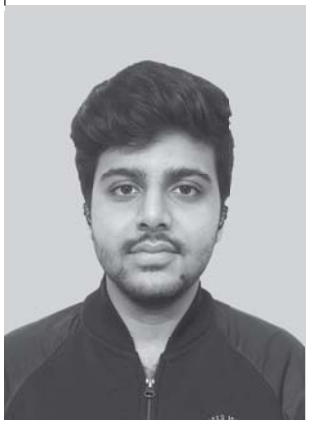
СЕРГЕЕВ Михаил Борисович

Профессор, директор Института вычислительных систем и программирования, заведующий кафедрой вычислительных систем и сетей Санкт-Петербургского государственного университета аэрокосмического приборостроения, почетный работник высшего профессионального образования РФ.
В 1980 году окончил ЛЭТИ по специальности «Электронные вычислительные машины».
В 2001 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 100 научных публикаций и 14 патентов на изобретения.
Область научных интересов — теория разрядных вычислений, методы проектирования спецпроцессоров для систем контроля и управления и др.
Эл. адрес: mbse@mail.ru

УШАКОВ Игорь Александрович

Доцент кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича.
В 2010 году окончил Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича по специальности «Защищенные системы связи».
В 2020 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 100 научных публикаций.
Область научных интересов — информационная безопасность, телекоммуникационные системы.
Эл. адрес: ushakovia@gmail.com

ШАРМА Яш Джитендер



Студент Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».
Является автором трех научных публикаций.
Область научных интересов — информационная безопасность, исследование социальных сетей.
Эл. адрес: yash3498@gmail.com

СОКОЛ Даниил Сергеевич

Оператор роты (научной) Военной академии связи им. Маршала Советского Союза С. М. Буденного, Санкт-Петербург.
В 2020 году окончил Донской государственный технический университет по специальности «Информационные системы в сфере защиты информации».
Является автором 39 научных публикаций и четырех программ для ЭВМ.
Область научных интересов — компьютерная безопасность, информационная безопасность, способы контроля уязвимостей и управления безопасностью информационно-вычислительных, информационно-телекоммуникационных и распределенных информационных сетей.
Эл. адрес: sokol_infosec@mail.ru

ФЕДОРЧЕНКО Елена Владимировна

Старший научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского института информатики и автоматизации РАН.
В 2009 году окончила с отличием Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» по специальности «Компьютерная безопасность».
В 2017 году защитила диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 70 научных публикаций.
Область научных интересов — безопасность компьютерных сетей, в том числе анализ защищенности и поддержка принятия решений по реагированию на кибератаки, интеллектуальный анализ данных.
Эл. адрес: doynikova@comsec.spb.ru

ШЕВЧЕНКО Александр Александрович

Научный сотрудник научно-исследовательского центра Военной академии связи им. Маршала Советского Союза С. М. Буденного, Санкт-Петербург.
В 2015 году окончил Рязанский государственный радиотехнический университет по специальности «Компьютерная безопасность».
Является автором 42 научных публикаций.
Область научных интересов — компьютерная безопасность, информационная безопасность, способы контроля уязвимостей и управления безопасностью информационно-вычислительных, информационно-телекоммуникационных и распределенных информационных сетей.
Эл. адрес: alex_pavel1991@mail.ru