Журнал зарегистрирован 01.03.2021 Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство о регистрации ПИ № ФС 77-80320

Адрес издателя и редакции: 630073, г. Новосибирск, пр. К. Маркса, 20.
E-mail: office@publish.nstu.ru и digital-tech-security@mail.ru
Web site: http://publish.nstu.ru и http://journals.nstu.ru/digital-tech-security/

Publisher and editorial office adress: 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation

До номера 1 (100) 2021 г. включительно журнал выходил под названием «Сборник научных трудов НГТУ» (ISSN 2307-6879)

16+

БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ

ISSN 2782-2230

№ 3 (114) **2024**

СОДЕРЖАНИЕ

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Выпускающий редактор *И.П. Брованова*
Корректор *Л.Н. Киншт*
Компьютерная верстка *С.И. Ткачева*

Лицензия № ИД 04303 от 20.03.01. Подписано в печать 26.09.2024. Выход в свет 27.09.2024
Формат 60×84 1/16. Бумага офсетная. Тираж 300 экз. Уч.-изд. л. 4,88
Печ. л. 5,25. Изд. № 118. Заказ № 179. Цена свободная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20

Publisher and editorial adress: 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation
E-mail: office@publish.nstu.ru, digital-tech-security@mail.ru
Web site: http://publish.nstu.ru, http://journals.nstu.ru/digital-tech-security/

DIGITAL TECHNOLOGY SECURITY

ISSN 2782-2230

№ 3 (114) **2024**

CONTENTS

METHODS AND SYSTEMS OF INFORMATION PROTECTION, INFORMATION SECURITY

Publishing Editor *I.P. Brovanova*
Editor *L.N. Kinsht*
Computer imposition *S.I. Tkacheva*

License № ID 04303 from 20.03.01. Signed in print September 26, 2024
Date of publication September 27, 2024. Format 60 × 84 1/16
Offset Paper. Circulation is 300 copies. Educational-ed. liter. 4,88. printed pages 5,25
Publishing number 118. Order number 179

It is printed in printing house of Novosibirsk State Technical University
630073, Novosibirsk, 20 K. Marx Prospekt

БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2024. – № 3 (114). – 9–20

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056.53 DOI: 10.17212/2782-2230-2024-3-9-20

РАСПОЗНАВАНИЕ ФИШИНГОВЫХ ССЫЛОК С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ*

К.А. ЛУКМАНОВА[1], В.М. КАРТАК[2]

[1] *450076, РФ, г. Уфа, ул. Заки Валиди, 32, Уфимский университет науки и технологий, аспирант кафедры вычислительной техники и защиты информации. E-mail: lukmanova.ka@gmail.ru*
[2] *450076, РФ, г. Уфа, ул. Заки Валиди, 32, Уфимский университет науки и технологий, доктор физико-математических наук, профессор, заведующий кафедрой вычислительной техники и защиты информации. E-mail: kartak.vm@ugatu.su*

В последние годы фишинг стал одной из наиболее распространенных и опасных киберугроз. Эти атаки направлены на получение конфиденциальной информации пользователей, такой как пароли и данные банковских карт, посредством обманных сообщений или веб-сайтов, что делает проблему защиты от них актуальной как никогда. Традиционные методы защиты от фишинга, такие как черные списки и эвристический анализ, уже не справляются с темпами эволюции фишинговых атак. В связи с этим возникает необходимость в разработке более современных и интеллектуальных методов, среди которых особое место занимают методы машинного обучения. В настоящей статье рассматриваются различные методы машинного обучения, применяемые для автоматического выявления фишинговых URL. В работе представлены основные подходы, архитектуры моделей, преимущества и недостатки каждого метода, а также проведен сравнительный анализ их эффективности на реальных данных.

Ключевые слова: фишинг, машинное обучение, фишинговые URL, киберугрозы, глубокое обучение, сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), логистическая регрессия, градиентный бустинг, случайный лес, классификация, кибербезопасность, анализ сетевого трафика, анализ веб-страниц, вычислительная сложность

* Статья получена 12 августа 2024 г.

ВВЕДЕНИЕ

Фишинг – это вид кибератаки, при которой злоумышленник пытается получить конфиденциальную информацию пользователей, такую как пароли, данные банковских карт или другая личная информация, выдавая себя за доверенное лицо или организацию. Чаще всего это происходит через электронные письма или сообщения, которые содержат ссылки на поддельные веб-сайты, визуально напоминающие настоящие. Эти фальшивые сайты обычно очень похожи на легитимные, что делает их трудноразличимыми для большинства пользователей [1].

Значительное увеличение числа фишинговых атак за последние годы связано с ростом интернет-коммерции и активным использованием цифровых сервисов. Согласно отчетам по информационной безопасности фишинг является одной из наиболее распространенных форм кибератак и продолжает совершенствоваться, адаптируясь к современным защитным мерам [2]. В результате традиционные методы защиты, такие как фильтрация по черным спискам, эвристические анализаторы, сигнатуры и ручной контроль, теряют свою эффективность [3].

Одним из решений этой проблемы является применение методов машинного обучения, которые способны автоматически распознавать и классифицировать фишинговые URL на основе анализа их характеристик. Такие методы могут не только повысить точность и скорость обнаружения фишинговых ссылок, но и обеспечить устойчивость к новым, ранее неизвестным атакам.

1. КЛАССИФИКАЦИЯ МЕТОДОВ РАСПОЗНАВАНИЯ ФИШИНГОВЫХ ССЫЛОК

Для эффективного распознавания фишинговых ссылок используются различные методы машинного обучения. Все они условно делятся на несколько категорий: методы анализа содержимого URL, методы анализа содержимого веб-страницы, методы анализа сетевого трафика и гибридные методы [4].

1.1. МЕТОДЫ НА ОСНОВЕ АНАЛИЗА СОДЕРЖИМОГО URL

Эти методы направлены на анализ характеристик URL, не углубляясь в содержимое веб-страницы. Они используют следующие признаки.

- Длина URL. Фишинговые ссылки часто имеют либо чрезмерно длинные, либо короткие URL.

- Использование поддоменов. Часто фишинговые сайты используют множество поддоменов для маскировки.
- Наличие подозрительных слов. В URL могут содержаться такие слова, как login, secure, verify, что может служить индикатором фишинга.
- Специальные символы. Например, наличие в URL символов “-”, “@”, “%”, которые часто используются для обмана пользователей.

Эти признаки могут быть использованы в качестве входных данных для различных моделей машинного обучения, таких как логистическая регрессия, деревья решений или случайный лес. Эти модели обучаются на данных, содержащих как легитимные, так и фишинговые URL, что позволяет им выявлять закономерности и строить классификаторы, способные разделять URL на безопасные и подозрительные.

Пример исследования. В одном из исследований была использована модель логистической регрессии, обученная на наборе данных, включающем 30 000 URL. В качестве признаков использовались длина URL, количество поддоменов, наличие подозрительных слов и другие характеристики. Результаты показали, что такая модель способна с высокой точностью (около 90 %) распознавать фишинговые URL [5].

Однако основной недостаток такого подхода заключается в его ограниченности – он может не справляться с новыми видами фишинга, которые используют новые, ранее не встречавшиеся техники маскировки.

1.2. МЕТОДЫ НА ОСНОВЕ АНАЛИЗА СОДЕРЖИМОГО ВЕБ-СТРАНИЦЫ

В отличие от методов, основанных на анализе URL, этот подход требует более глубокого анализа содержимого веб-страницы. Он включает следующие аспекты.

- SSL-сертификат. Проверка наличия и подлинности SSL-сертификата, который является признаком защищенности веб-сайта.
- Соответствие доменного имени содержимому страницы. Например, если доменное имя не соответствует тематике или содержимому сайта, это может быть признаком фишинга.
- Количество и типы внешних ссылок. Фишинговые сайты часто содержат множество внешних ссылок на подозрительные ресурсы.
- Анализ текста страницы. Автоматический анализ текста на наличие ошибок, мошеннических предложений и т. д.

Эти методы требуют значительно большего количества вычислительных ресурсов по сравнению с анализом URL. Однако они позволяют выявить фишинг с высокой точностью за счет анализа контекста страницы [6, 7].

Пример исследования. В исследовании, проведенном группой ученых, использовались случайные леса и градиентный бустинг для классификации фишинговых сайтов. Такие модели обучались на большом наборе данных, содержащем как фишинговые, так и легитимные сайты. В качестве признаков использовались различные характеристики страниц, такие как наличие SSL-сертификата, метаданные и внешний вид сайта. Результаты показали, что эти методы позволяют достичь точности до 95 %, что делает их весьма эффективными для обнаружения фишинговых сайтов. Однако такой подход требует значительных вычислительных ресурсов и времени на анализ каждой страницы, и это может ограничивать его применение в реальном времени.

1.3. МЕТОДЫ НА ОСНОВЕ АНАЛИЗА ТРАФИКА

Методы анализа трафика предполагают изучение поведения пользователя и характеристик взаимодействий с веб-сайтами. Они учитывают следующие аспекты.

- Частота посещений домена. Анализируется, как часто и кем посещается данный домен.
- Время нахождения на сайте. Временны́е характеристики сессий могут свидетельствовать о ненадежности сайта.
- Повторные переходы. Повторные переходы по подозрительным ссылкам могут служить признаком фишинга.

На основе этих данных строятся поведенческие модели, которые могут выявлять аномалии, характерные для фишинговых атак. Применение методов кластеризации, таких как K-средние, позволяет группировать схожие по поведению URL и выявлять потенциально опасные.

Пример исследования. Одно из исследований изучало использование методов кластеризации для обнаружения фишинговых сайтов на основе анализа сетевого трафика. Исследователи использовали метод K-средних для классификации URL, основываясь на таких признаках, как частота посещений и время, проведенное на сайте. Результаты показали, что с помощью такого метода можно с точностью до 85 % различать фишинговые и легитимные сайты. Однако его основным недостатком является зависимость от наличия достаточного объема данных о поведении пользователей, что может ограничивать его применение на новых или редко посещаемых сайтах [8].

1.4. ГИБРИДНЫЕ МЕТОДЫ

Гибридные методы представляют собой комбинацию нескольких подходов с целью повышения точности и надежности распознавания фишинговых

ссылок. Например, можно объединить анализ URL и контент-ориентированный подход для получения более детальной информации о ссылке и странице. Это позволяет учитывать как поверхностные, так и более глубокие признаки [9].

Сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN) позволяют автоматически извлекать сложные признаки и комбинировать их для классификации. CNN хорошо подходят для анализа текстовых данных, таких как URL и текст страницы, тогда как RNN могут учитывать временны́е зависимости и историю взаимодействия с сайтом.

Пример исследования. В исследовании была предложена гибридная модель, основанная на комбинации сверточной нейронной сети для анализа URL и рекуррентной нейронной сети для анализа содержимого веб-страницы и поведения пользователя. Эта модель была обучена на большом наборе данных и показала точность распознавания фишинговых сайтов свыше 97 %. Это значительно превышает точность традиционных методов, однако требует значительных вычислительных ресурсов. Основным преимуществом такой модели является ее способность адаптироваться к новым видам фишинга и быстро обучаться на новых данных.

2. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Для оценки эффективности различных методов машинного обучения был проведен эксперимент на большом наборе данных, включающем как фишинговые, так и легитимные URL. В созданном нами наборе данных, который был загружен в систему, общее количество примеров сайтов с легитимными ссылками 1094, а количество примеров, относящихся к фишинговым URL, – 1362. Все признаки нормализованы и имеют бинарные значения для определения: от –1 до 1, где –1 означает фишинговую ссылку, 0 означает подозрительную ссылку, и 1 означает легитимную ссылку. Нулевой признак подозрительной ссылки показывает, что веб-страница может быть или фишинговой, или настоящей, т. е. ссылка содержит в себе как некоторые «законные», так и фальшивые признаки.

При помощи функции языка Python TF-IDF разделили набор данных на тестовую и обучающую выборку. Для этого использовали соотношение 20 % и 80 %. В полученной обучающей выборке содержится 1081 фишинговая запись и 883 «законные». Остальная часть отправлена в тестовую выборку.

Были протестированы следующие модели: логистическая регрессия, случайный лес, градиентный бустинг, сверточная нейронная сеть и рекуррентная нейронная сеть. Основными метриками для оценки качества моделей были

выбраны точность (accuracy), полнота (recall), F-мера (F1-score) и площадь под кривой ROC (ROC-AUC).

2.1. ЛОГИЧЕСКАЯ РЕГРЕССИЯ

Логистическая регрессия – один из простейших методов классификации, который показал себя достаточно эффективным в задачах, связанных с бинарной классификацией, таких как распознавание фишинговых URL. В проведенном эксперименте логистическая регрессия продемонстрировала средний уровень точности на уровне 88 %. Это связано с тем, что данный метод ограничен линейностью используемой модели, что не позволяет учитывать более сложные и нелинейные зависимости между признаками.

2.2. СЛУЧАЙНЫЙ ЛЕС

Случайный лес (Random Forest) – это ансамблевый метод, который использует множество деревьев решений для улучшения устойчивости и точности классификации. В эксперименте случайный лес показал более высокие результаты по сравнению с логистической регрессией, достигая точности около 92 %. Этот метод особенно хорошо справляется с задачами, в которых необходимо учитывать взаимодействие между большим числом признаков. Однако его основным недостатком является увеличение сложности модели и соответственно времени на ее обучение и предсказание.

2.3. ГРАДИЕНТНЫЙ БУСТИНГ

Градиентный бустинг – еще один ансамблевый метод, который использует последовательное построение деревьев решений с целью минимизации ошибок предыдущих моделей. В эксперименте этот метод показал одну из наилучших точностей среди классических методов машинного обучения, достигая 94 %. Градиентный бустинг особенно эффективен при наличии большого объема данных и большого числа признаков. Однако, подобно случайному лесу, он требует значительных вычислительных ресурсов, особенно при обработке больших наборов данных.

2.4. СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ

Сверточные нейронные сети (CNN) изначально разработаны для обработки данных с локальными зависимостями, таких как изображения, однако

они также оказались эффективными и для текстовых данных, таких как URL. В эксперименте CNN продемонстрировала высокую точность, превышающую 95 %. Это связано с ее способностью извлекать сложные признаки, которые могут быть неочевидны для классических методов машинного обучения. CNN особенно полезны для распознавания паттернов в тексте URL, что делает их мощным инструментом в задачах классификации фишинговых ссылок.

2.5. РЕКУРРЕНТНАЯ НЕЙРОННАЯ СЕТЬ

Рекуррентные нейронные сети (RNN) и их более современные версии, такие как LSTM (Long Short-Term Memory), используются для анализа последовательных данных и временны́х рядов. В контексте фишинговых ссылок они могут анализировать последовательности символов в URL или даже последовательности действий пользователя на сайте. В эксперименте RNN показала точность около 93 %, уступая CNN, но превосходя традиционные методы машинного обучения. Основное преимущество RNN заключается в ее способности учитывать контекст и историю взаимодействий, что может быть полезно для обнаружения более сложных фишинговых атак.

3. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ

Точность и надежность. Анализ показал, что современные методы, основанные на глубоком обучении (CNN и RNN), демонстрируют наивысшую точность и надежность при распознавании фишинговых URL. Эти методы способны автоматически извлекать сложные признаки и учитывать нелинейные зависимости, что делает их более эффективными в условиях динамически меняющихся атак. Однако их использование связано с высокими вычислительными затратами, что может ограничивать их применение в условиях реального времени или на устройствах с ограниченными ресурсами.

Классические методы машинного обучения, такие как логистическая регрессия и деревья решений, показали более низкую точность, однако они имеют преимущество в скорости работы и простоте реализации. Эти методы могут быть применимы в ситуациях, когда необходимо быстрое и простое решение, не требующее значительных вычислительных ресурсов [10].

Гибкость и адаптивность. Гибридные методы, такие как комбинация CNN и RNN, показали наилучшие результаты с точки зрения адаптивности к новым атакам. Такие модели могут быть быстро переобучены на новых данных, что позволяет им эффективно справляться с новыми видами фишинго-

вых атак. Это особенно важно в условиях, когда злоумышленники постоянно адаптируют свои методы для обхода традиционных защитных мер.

Классические методы, такие как градиентный бустинг и случайный лес, также обладают определенной гибкостью, однако они требуют более частого обновления моделей и не всегда могут эффективно обрабатывать новые виды атак без значительных изменений в архитектуре модели [11].

Вычислительная сложность. Одним из ключевых факторов при выборе метода является вычислительная сложность. Методы глубокого обучения, такие как CNN и RNN, требуют значительных ресурсов для обучения и предсказания, что может ограничивать их применение в условиях реального времени. С другой стороны, классические методы, такие как логистическая регрессия и случайный лес, являются менее ресурсоемкими и могут быть использованы для быстрых предсказаний на больших объемах данных.

ЗАКЛЮЧЕНИЕ

В настоящей статье были рассмотрены различные методы машинного обучения, используемые для распознавания фишинговых URL. Мы провели сравнительный анализ нескольких моделей, включая как классические методы машинного обучения, так и современные подходы на основе глубокого обучения. Результаты показали, что гибридные методы, такие как комбинация CNN и RNN, обладают наивысшей точностью и адаптивностью. Это делает их наиболее перспективными для использования в условиях постоянно меняющейся среды киберугроз.

Однако использование глубоких нейронных сетей связано с высокими вычислительными затратами, что может ограничивать их применение в реальных условиях. В то же время более простые модели, такие как логистическая регрессия и деревья решений, обеспечивают приемлемый уровень точности при низких затратах на вычисления, что делает их подходящими для быстрого фильтрационного анализа.

В будущем целесообразно развивать методы, направленные на уменьшение вычислительной сложности моделей глубокого обучения, а также на интеграцию различных подходов для повышения общей эффективности распознавания фишинговых атак. Кроме того, важным направлением дальнейших исследований является разработка методов, способных эффективно адаптироваться к новым и неизвестным типам атак, что позволит значительно повысить уровень информационной безопасности в сети.

СПИСОК ЛИТЕРАТУРЫ

1. *Карпова Н.Е., Восканян И.И.* Угроза социальной инженерии и фишинга в современной информационной безопасности // Безопасность цифровых технологий. – 2024. – № 2 (113). – С. 69–78. – DOI: 10.17212/2782-2230-2024-2-69-78.

2. Phishing Attack Trends Report – 4Q 2023 / APWG. Phishing Activity Trends Reports. – URL: https://apwg.org/trendsreports/ (accessed 28.08.2024).

3. *Hussein S.K., Wahaballah A., Alosaimi A.* Detecting phishing websites using natural language processing // International Journal of Computer Engineering in Research Trends. – 2021. – Vol. 8 (12). – P. 220–227.

4. *Кутлыев Д.З., Шманина А.В.* Использование алгоритмов машинного обучения для защиты от URL-фишинга // Мавлютовские чтения: XV Всероссийская молодежная научная конференция. – Уфа, 2021. – Т. 4. – С. 430–435.

5. Classifying phishing URLs using recurrent neural networks / A.C. Bahnsen, I.D. Torroledo, J. Camacho, S. Villegas // 2017 APWG Symposium on Electronic Crime Research (eCrime). – IEEE, 2017. – DOI: 10.1109/ECRIME.2017.7945048.

6. Machine learning based phishing detection from URLs / O.K. Sahingoz, E. Buber, O. Demir, B. Diri // Expert Systems with Applications. – 2019. – Vol. 117. – P. 345–357.

7. *Лукманова К.А., Картак В.М.* Векторное представление слов в задаче анализа текстовых сообщений // Мавлютовские чтения: XIV Всероссийская молодежная научная конференция. – Уфа, 2020. – Т. 5, ч. 2. – Ст. 25.

8. *Артюшкина Е.С., Андирякова О.О., Тюрина Д.А.* Использование методов машинного обучения при анализе сетевого трафика и вредоносного программного обеспечения // Индустриальная экономика. – 2023. – № 4. – С. 12–15. – DOI: 10.47576/2949-1886_2023_4_12.

9. *Мухамадиева К.Б., Муминов Б.Б.* Обзор методов обнаружения фишинговых атак на основе искусственного интеллекта // Вестник Донецкого национального университета. Серия Г: Технические науки. – 2021. – № 4. – С. 37–45.

10. Beyond blacklists: learning to detect malicious web sites from suspicious URLs / J. Ma, L.K. Saul, S. Savage, G.M. Voelker // KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – ACM, 2009. – P. 1245–1254. – DOI: 10.1145/1557019.1557153.

11. *Dutta A.K.* Detecting phishing websites using machine learning technique // PLoS ONE. – 2021. – Vol. 16 (10). – P. e0258361. – DOI: 10.1371/journal.pone.0258361.

Лукманова Карина Александровна, аспирант кафедры вычислительной техники и защиты информации Уфимского университета науки и технологий. Основное направление научных исследований – информационная безопасность, машинное обучение. E-mail: lukmanova.ka@gmail.ru

Картак Вадим Михайлович, доктор физико-математических наук, заведующий кафедрой вычислительной техники и защиты информации Уфимского университета науки и технологий. Область научных интересов – информационная безопасность, дискретная оптимизация. E-mail: kartak.vm@ugatu.su

DOI: 10.17212/2782-2230-2024-3-9-20

Recognition of phishing links using machine learning methods*

K.A. Lukmanova[1], V.M. Kartak[2]

[1] *Ufa University of Science and Technology, 32 Zaki Validi Street, Ufa, 450076, Russian Federation, postgraduate student at the Department of Computer Engineering and Information Security. E-mail: lukmanova.ka@gmail.ru*

[2] *Ufa University of Science and Technology, 32 Zaki Validi Street, Ufa, 450076, Russian Federation, Doctor of Physical and Mathematical Sciences, Professor, head of the Computer Engineering and Information Security Department. E-mail: kartak.vm@ugatu.su*

In recent years, phishing has become one of the most widespread and dangerous cyber threats. These attacks aim to obtain users' confidential information, such as passwords and credit card details, through deceptive messages or websites, making the issue of protection against them more relevant than ever. Traditional methods of phishing protection, such as blacklists and heuristic analysis, can no longer keep up with the evolving pace of phishing attacks. Therefore, there is a need to develop more advanced and intelligent methods, among which machine learning (ML) techniques play a significant role. This article discusses various ML methods used for automatic detection of phishing URLs. The study presents the main approaches, model architectures, advantages and disadvantages of each method, and provides a comparative analysis of their effectiveness on real data.

Keywords: phishing, machine learning, phishing URLs, cyber threats, deep learning, convolutional neural networks (CNN), recurrent neural networks (RNN), logistic regression, gradient boosting, random forest, classification, cybersecurity, network traffic analysis, web page analysis, computational complexity

* *Received 12 August 2024.*

REFERENCES

1. Karpova N.E., Voskanyan I.I. Ugroza sotsial'noĭ inzhenerii i fishinga v sovremennoĭ informatsionnoĭ bezopasnosti [Threat of social engineering and phishing in modern information security]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 2 (113), pp. 69–78. DOI: 10.17212/2782-2230-2024-2-69-78.

2. APWG. *Phishing Attack Trends Report – 4Q 2023*. Available at: https://apwg.org/trendsreports/ (accessed 28.08.2024).

3. Hussein S.K., Wahaballah A., Alosaimi A. Detecting phishing websites using natural language processing. *International Journal of Computer Engineering in Research Trends*, 2021, vol. 8 (12), pp. 220–227.

4. Kutlyev D.Z., Shmanina A.V. [Using machine learning algorithms to protect against URL phishing]. *Mavlyutovskie chteniya: XV Vserossiiskaya molodezhnaya nauchnaya konferentsiya* [Mavlyutov Readings. Proceedings of the XV All-Russian Youth Scientific Conference]. Ufa, 2021, vol. 4, pp. 430–435. (In Russian).

5. Bahnsen A.C., Torroledo I.D., Camacho J., Villegas S. Classifying phishing URLs using recurrent neural networks. *2017 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2017. DOI: 10.1109/ECRIME.2017.7945048.

6. Sahingoz O.K., Buber E., Demir O., Diri B. Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 2019, vol. 117, pp. 345–357.

7. Lukmanova K.A., Kartak V.M. [Vector representation of words in the task of text message analysis]. *Mavlyutovskie chteniya: XIV Vserossiiskaya molodezhnaya nauchnaya konferentsiya* [Mavlyutov Readings. Proceedings of the XIV All-Russian Youth Scientific Conference]. Ufa, 2020, vol. 5, pt. 2, art. 25. (In Russian).

8. Artyushkina E.S., Andiryakova O.O., Tyurina D.A. Ispol'zovanie metodov mashinnogo obucheniya pri analize setevogo trafika i vredonosnogo programmnogo obespecheniya [Using machine learning methods in analyzing network traffic and malicious software]. *Industrial'naya ekonomika = Industrial Economics*, 2023, no. 4, pp. 12–15. DOI: 10.47576/2949-1886_2023_4_12.

9. Mukhamadieva K.B., Muminov B.B. Obzor metodov obnaruzheniya fishingovykh atak na osnove iskusstvennogo intellekta [Review of phishing attack detection methods based on artificial intelligence]. *Vestnik Donetskogo natsional'nogo universiteta. Seriya G: Tekhnicheskie nauki = Bulletin of Donetsk National University. Series G: Technical Sciences*, 2021, no. 4, pp. 37–45.

10. Ma J., Saul L.K., Savage S., Voelker G.M. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. *KDD '09: Proceedings of the*

15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2009, pp. 1245–1254. DOI: 10.1145/1557019.1557153.

11. Dutta A.K. Detecting phishing websites using machine learning technique. *PLoS ONE*, 2021, vol. 16 (10), p. e0258361. DOI: 10.1371/journal.pone.0258361.

Для цитирования:

Лукманова К.А., Картак В.М. Распознавание фишинговых ссылок с использованием методов машинного обучения // Безопасность цифровых технологий. – 2024. – № 3 (114). – С. 9–20. – DOI: 10.17212/2782-2230-2024-3-9-20.

For citation:

Lukmanova K.A., Kartak V.M. Raspoznavanie fishingovykh ssylok s ispol'zovaniem metodov mashinnogo obucheniya [Recognition of phishing links using machine learning methods]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 3 (114), pp. 9–20. DOI: 10.17212/2782-2230-2024-3-9-20.

БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2024. – № 3 (114). – 21–33

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056.5 DOI: 10.17212/2782-2230-2024-3-21-33

МУЛЬТИАГЕНТНОЕ ТЕСТИРОВАНИЕ НА ПРОНИКНОВЕНИЕ НА ОСНОВЕ AIRL*

М.М. ГРЕКОВ[1], А.А. СЫЧУГОВ[2]

[1] *300012, РФ, г. Тула, пр. Ленина, 92, Тульский государственный университет, ассистент кафедры информационной безопасности. E-mail: grekov.web@yandex.ru*
[2] *300012, РФ, г. Тула, пр. Ленина, 92, Тульский государственный университет, заведующий кафедрой информационной безопасности. E-mail: xru2003@list.ru*

В статье рассматривается применение мультиагентного подхода на основе метода Adversarial Inverse Reinforcement Learning (AIRL) для тестирования на проникновение в информационные системы. Описаны теоретические аспекты мультиагентного AIRL, включая возможности моделирования сложных и многоступенчатых атак, координации действий агентов, а также обучения с частичным наблюдением, что позволяет учитывать ограничения в доступе к информации. Практическое применение такого подхода продемонстрирует его эффективность в выявлении уязвимостей, обеспечивая более глубокий и точный анализ безопасности.

Ключевые слова: состязательное обучение с обратным подкреплением, информационная безопасность, тестирование на проникновение, автоматизация, мультиагентное обучение, частичное наблюдение, машинное обучение, нейронные сети

ВВЕДЕНИЕ

Информационная безопасность становится одной из ключевых задач в современном мире, где данные и цифровые ресурсы играют критическую роль в функционировании организаций, правительств и общества в целом. Угроза кибератак растет с каждым годом, и злоумышленники продолжают разрабатывать всё более сложные и изощренные методы компрометации систем. В связи с этим обеспечение безопасности информационных систем требует использования продвинутых методов и подходов, которые способны эффективно выявлять и устранять уязвимости до того, как они будут использованы в атаках.

* Статья получена 10 августа 2024 г.

Тестирование на проникновение (Penetration Testing, PT) – один из важнейших инструментов обеспечения информационной безопасности, направленный на обнаружение слабых мест в системах путем имитации реальных атак. Однако традиционные методы тестирования на проникновение сталкиваются с рядом ограничений. Во-первых, они зачастую предполагают сценарии атак, которые не учитывают сложность и динамичность современных кибератак, включающих координацию между несколькими участниками или использованием разнообразных тактик и стратегий. Во-вторых, традиционные методы требуют значительных затрат ресурсов и времени, особенно при тестировании крупных и комплексных сетевых инфраструктур. Эти ограничения требуют разработки новых методов и подходов, которые позволят повысить эффективность и глубину анализа систем безопасности.

Одним из перспективных направлений, способных решить указанные проблемы, является применение нейросетевых алгоритмов и методов машинного обучения. В частности, применение метода обратного подкрепления (Adversarial Inverse Reinforcement Learning, AIRL), который позволяет агентам обучаться на основе демонстраций, создавая оптимальные стратегии для достижения заданных целей [1, 2]. Мультиагентный подход в AIRL представляет собой дальнейшее развитие этого метода, при котором несколько агентов взаимодействуют друг с другом и с окружающей средой для моделирования сложных сценариев атак и тестирования на проникновение [3].

Таким образом, мультиагентный подход в AIRL представляет собой мощный и гибкий инструмент для тестирования на проникновение, который позволяет моделировать сложные и многоступенчатые атаки, координировать действия агентов, обучать их на основе частичной информации и использовать ансамбли агентов для повышения эффективности анализа. Этот подход имеет потенциал значительно улучшить безопасность информационных систем, позволяя обнаруживать и устранять уязвимости на более ранних этапах и с большей точностью. В настоящей статье рассматриваются теоретические аспекты мультиагентного AIRL, а также его практическое применение для тестирования на проникновение, что позволяет расширить возможности такого тестирования и сделать его более адаптивным, масштабируемым и эффективным в условиях современных угроз.

1. МУЛЬТИАГЕНТНЫЙ ПОДХОД AIRL

Применение мультиагентного AIRL для тестирования на проникновение открывает новые горизонты для анализа и выявления уязвимостей в информационных системах. Такой подход позволяет моделировать много-

ступенчатые атаки, в которых действия одного агента зависят от решений и поведения других агентов. Это особенно важно в условиях, когда злоумышленники координируют свои действия для преодоления защитных мер и достижения своих целей. Мультиагентные системы также способны адаптироваться к изменяющимся условиям, что позволяет им находить уязвимости, которые могут быть упущены при использовании традиционных методов.

Важной особенностью мультиагентного AIRL является возможность моделирования как кооперативных, так и конкурентных сценариев, что делает его особенно полезным для тестирования на проникновение. Кооперативные сценарии могут включать взаимодействие агентов для выявления и использования уязвимостей, тогда как конкурентные сценарии позволяют моделировать противодействие между несколькими атакующими агентами для формирования наиболее эффективных траекторий атак.

Многоуровневое обучение, являющееся одной из ключевых составляющих мультиагентного AIRL, позволяет разбивать сложные задачи на несколько уровней абстракции, что упрощает координацию действий агентов и позволяет более эффективно решать сложные задачи. Это особенно актуально в условиях тестирования на проникновение, когда необходимо моделировать атаки на различных уровнях системы – от сети до приложений и данных.

Еще одной важной характеристикой мультиагентного AIRL является возможность обучения с частичным наблюдением, что позволяет агентам принимать решения на основе неполной или неточной информации о системе. Это приближает модель к реальным условиям тестирования на проникновение, когда атакующие не всегда имеют полный доступ к информации о цели и должны действовать на основе ограниченных данных.

2. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

2.1. ФОРМАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ МЕЖДУ НЕСКОЛЬКИМИ АГЕНТАМИ

Мультиагентный подход в Adversarial Inverse Reinforcement Learning (AIRL) является расширением классического алгоритма AIRL, при котором вместо одного агента существует несколько агентов, взаимодействующих друг с другом и с окружающей средой. Такой подход особенно важен для моделирования сложных систем, в которых поведение одного агента зависит от действий других агентов.

В мультиагентных системах каждый агент принимает решения исходя из текущего состояния среды и действий других агентов. Это взаимодействие может быть для кооперативных, конкурентных или смешанных систем.

- Кооперативные системы. Агенты работают вместе для достижения общей цели. Примеры включают командные виды спорта и координированные задачи в робототехнике.
- Конкурентные системы. Агенты преследуют противоположные цели. Примеры включают игры, такие как шахматы или го, где выигрыш одного агента означает проигрыш другого.
- Смешанные системы. Содержат элементы как кооперации, так и конкуренции. Например, экономические модели, когда агенты могут сотрудничать для достижения некоторых целей, но конкурируют за ресурсы.

В мультиагентном AIRL каждый агент обучает свою политику, учитывая действия и стратегии других агентов. Это требует более сложного подхода к обучению, так как агенты должны адаптироваться к изменяющимся стратегиям своих коллег.

Каждый агент имеет свой дискриминатор, который помогает оценивать, насколько действия агента соответствуют демонстрациям. Дискриминаторы также помогают агентам различать истинные демонстрации от сгенерированных траекторий, способствуя обучению более реалистичных и эффективных стратегий.

Состояния и действия:

- S – множество всех возможных состояний среды;
- A_i – множество действий агента i;
- $A = (A_1, A_2, \ldots, A_n)$ – совместное множество действий всех агентов.

В мультиагентной системе состояние среды $s \in S$ является общим для всех агентов, но каждое действие $a_i \in A_i$ выполняется отдельным агентом i. Совместное действие всех агентов представляется вектором

$$a = (a_1, a_2, \ldots, a_n). \quad (1)$$

Каждый агент i имеет свою собственную политику $\pi_{\theta i}(a_i | s)$, которая определяет выбор действия a_i в зависимости от текущего состояния s. Политики всех агентов вместе образуют совокупную политику системы

$$\Pi = \{\pi_{\theta_1}, \pi_{\theta_2}, \ldots, \pi_{\theta_n}\}. \quad (2)$$

$P(s' \mid s, a)$ – функция перехода состояний, определяющая вероятность перехода из состояния s в состояние s' при выполнении совместного действия $a = (a_1, a_2, \ldots, a_n)$.

В мультиагентной системе каждый агент i имеет следующие компоненты.

1. Политика $\pi_{\theta_i}(a_i \mid s)$. Определяет вероятности выбора действия a_i в состоянии s для агента i. Параметризована вектором θ_i.

2. Функция вознаграждений $R_i(s, a_i, a_{-i})$. Вознаграждение агента i в состоянии s за действие a_i, учитывающее действия остальных агентов $a_{-i} = (a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$.

3. Дискриминатор $D_{\phi_i}(s, a_i)$. Оценивает вероятность того, что пара «состояние, действие» (s, a_i) агента i взята из истинных демонстраций.

Дискриминатор агента i обучается различать истинные демонстрации и траектории, сгенерированные текущей политикой агента:

$$D_{\phi_i}(s, a_i) = \frac{\exp\left(f_{\phi_i}(s, a_i)\right)}{\exp\left(f_{\phi_i}(s, a_i)\right) + \pi_{\theta_i}(a_i \mid s)}. \tag{3}$$

Функция потерь для дискриминатора:

$$L_{D_{\phi_i}} = E_{(s,a_i)\sim X_i}\left[\log D_{\phi_i}(s, a_i)\right] + E_{(s,a_i)\sim \pi_{\theta_i}}\left[\log\left(1 - D_{\phi_i}(s, a_i)\right)\right]. \tag{4}$$

Политика агента i обучается минимизировать функцию потерь, чтобы действия агента соответствовали истинным демонстрациям.

Функция вознаграждения для агента i:

$$R_i(s, a_i) = f_{\phi_i}(s, a_i) - \log \pi_{\theta_i}(a_i \mid s). \tag{5}$$

Функция потерь для политики:

$$L_G(\theta_i) = E_{\tau_i \sim \pi_{\theta_i}}\left[\sum_{t=0}^{T}\left(f_{\phi_i}(s_t, a_{i,t}) - \log \pi_{\theta_i}(a_{i,t} \mid s_t)\right)\right]. \tag{6}$$

Обновление параметров происходит итеративно.

1. Обновление дискриминатора:

$$\phi_i \leftarrow \phi_i + \eta_{\phi_i} \nabla_{\phi_i} L_{D_{\phi_i}} . \quad (7)$$

2. Обновление политики:

$$\theta_i \leftarrow \theta_i - \eta_{\theta_i} \nabla_{\theta_i} L_G(\theta_i) . \quad (8)$$

Мультиагентный подход в состязательном обучении с обратным подкреплением позволяет моделировать и обучать агентов, взаимодействующих друг с другом в сложных системах. Это расширяет возможности традиционного AIRL, делая его применимым к более широкому кругу задач, где поведение одного агента зависит от действий других агентов. В результате агенты могут обучаться более сложным стратегиям и взаимодействиям, что ведет к более реалистичному и эффективному поведению в реальных системах.

Применение мультиагентного подхода в контексте тестирования на проникновение может значительно усилить эффективность и глубину анализа безопасности сетевых систем. В этом контексте каждый агент может специализироваться на определенных аспектах безопасности, работая совместно для более комплексного и всестороннего тестирования.

2.2. ФОРМАЛИЗАЦИЯ МНОГОУРОВНЕВОГО ОБУЧЕНИЯ (HIERARCHICAL LEARNING)

Многоуровневое обучение (Hierarchical Learning) – это подход, который разбивает сложные задачи на несколько уровней абстракции или подзадач, что позволяет агентам решать их более эффективно. В контексте мультиагентного Adversarial Inverse Reinforcement Learning (AIRL) многоуровневое обучение может быть применено для улучшения координации и взаимодействия агентов [4, 5].

1. *Иерархические модели* делят задачу на высокоуровневые цели и низкоуровневые подзадачи. В мультиагентных системах это может означать разделение на стратегические (высокоуровневые) и тактические (низкоуровневые) задачи.

2. *Менеджеры и рабочие*. Высокоуровневые агенты (менеджеры) отвечают за определение целей и стратегий, в то время как низкоуровневые агенты (рабочие) выполняют конкретные действия для достижения этих целей.

В многоуровневом мультиагентном AIRL каждый агент имеет две политики:

- *высокоуровневая политика* $\pi_{\theta_i^H}(g \mid s)$: определяет цели или подзадачи g в состоянии s;
- *низкоуровневая политика* $\pi_{\theta_i^L}(a_i \mid s, g)$: определяет конкретные действия a_i для выполнения цели g в состоянии s.

Для каждого уровня политики существует свой дискриминатор:

- *дискриминатор высокоуровневой политики* $D_{\phi_i^H}(s, g)$: оценивает вероятность того, что цель g в состоянии s является частью истинных демонстраций;
- *дискриминатор низкоуровневой политики* $D_{\phi_i^L}(s, a_i \mid g)$: оценивает вероятность того, что действие a_i в состоянии s для цели g является частью истинных демонстраций.

Высокоуровневая политика:

$$\pi_{\theta_i^H}(g \mid s) = Pr\left(g \mid s;\ \theta_i^H\right). \tag{9}$$

Низкоуровневая политика:

$$\pi_{\theta_i^L}(a_i \mid s, g) = Pr\left(a_i \mid s, g;\ \theta_i^L\right). \tag{10}$$

Дискриминатор высокоуровневой политики:

$$D_{\phi_i^H}(s, g) = \frac{\exp\left(f_{\phi_i^H}(s, g)\right)}{\exp\left(f_{\phi_i^H}(s, g)\right) + \pi_{\theta_i^H}(g \mid s)}. \tag{11}$$

Дискриминатор низкоуровневой политики:

$$D_{\phi_i^L}(s, a_i \mid g) = \frac{\exp\left(f_{\phi_i^L}(s, a_i \mid g)\right)}{\exp\left(f_{\phi_i^L}(s, a_i \mid g)\right) + \pi_{\theta_i^L}(a_i \mid s, g)}. \tag{12}$$

Функция потерь для дискриминатора высокоуровневой политики:

$$L_{D_{\phi_i^H}} = E_{(s,g)\sim X_i}\left[\log D_{\phi_i^H}(s,g)\right] + E_{(s,g)\sim\pi_{\theta_i^H}}\left[\log\left(1 - D_{\phi_i^H}(s,g)\right)\right]. \quad (13)$$

Функция потерь для дискриминатора низкоуровневой политики:

$$L_{D_{\phi_i^L}} = E_{(s,a_i,g)\sim X_i}\left[\log D_{\phi_i^L}(s,a_i|\, g)\right] +$$

$$+ E_{(s,a_i,g)\sim\pi_{\theta_i^L}}\left[\log\left(1 - D_{\phi_i^L}(s,a_i|\, g)\right)\right]. \quad (14)$$

Функция вознаграждения для высокоуровневой политики:

$$R_i^H(s,g) = f_{\phi_i^H}(s,g) - \log\pi_{\theta_i^H}(g \mid s). \quad (15)$$

Функция потерь для высокоуровневой политики:

$$L_G^H\left(\theta_i^H\right) = E_{\tau_i\sim\pi_{\theta_i^H}}\left[\sum_{t=0}^{T}\left(f_{\phi_i^H}(s_t,g_t) - \log\pi_{\theta_i^H}(g_t|\, s_t)\right)\right]. \quad (16)$$

Функция вознаграждения для низкоуровневой политики:

$$R_i^L(s,a_i|\, g) = f_{\phi_i^L}(s,a_i|\, g) - \log\pi_{\theta_i^L}(a_i|\, s,g). \quad (17)$$

Функция потерь для низкоуровневой политики:

$$L_G^L\left(\theta_i^L\right) = E_{\tau_i\sim\pi_{\theta_i^L}}\left[\sum_{t=0}^{T}\left(f_{\phi_i^L}(s_t,a_{i,t}|\, g_t) - \log\pi_{\theta_i^L}(a_{i,t}|\, s_t,g_t)\right)\right]. \quad (18)$$

2.3. ФОРМАЛИЗАЦИЯ ОБУЧЕНИЯ С ЧАСТИЧНЫМ НАБЛЮДЕНИЕМ

Обучение с частичным наблюдением (Partially Observable Learning) относится к случаям, когда агенты не имеют полного доступа к состоянию среды и могут принимать решения только на основе частичной информации [6, 7]. В мультиагентных системах это добавляет дополнительный уровень слож-

ности, так как каждый агент должен принимать решения, учитывая как ограниченную информацию о среде, так и действия других агентов.

Частичные наблюдения o_i – наблюдения, которые доступны агенту *i*, зависят от истинного состояния среды *s* и могут содержать шум.

Политики агентов $\pi_{\theta_i}(a_i|\, o_i)$ – политики, которые агенты используют для принятия решений на основе своих наблюдений.

Дискриминаторы $D_{\phi_i}(o_i, a_i)$ – дискриминаторы, которые оценивают вероятность того, что пара «наблюдение, действие» является частью истинных демонстраций.

Агент *i* получает частичное наблюдение $o_i \in O_i$, которое связано с истинным состоянием среды $s \in S$ через наблюдательную модель *Z*:

$$o_i \sim Z(o_i|\, s)\,. \tag{19}$$

Политика агента *i* определяется как вероятностное распределение действий a_i на основе частичного наблюдения o_i:

$$\pi_{\theta_i}(a_i|\, o_i) = Pr(a_i|\, o_i;\ \theta_i)\,. \tag{20}$$

Дискриминатор агента *i* оценивает вероятность того, что пара «наблюдение, действие» $(a_i|\, o_i)$ взята из истинных демонстраций. Функция дискриминатора:

$$D_{\phi_i}(o_i, a_i) = \frac{\exp\left(f_{\phi_i}(o_i, a_i)\right)}{e\exp\left(f_{\phi_i}(o_i, a_i)\right) + \pi_{\theta_i}(a_i|\, o_i)}\,. \tag{21}$$

Функция потерь для дискриминатора:

$$L_{D_{\phi_i}} = E_{(o_i,a_i)\sim X_i}\left[\log D_{\phi_i}(o_i, a_i)\right] + E_{(o_i,a_i)\sim\pi_{\theta_i}}\left[\log\left(1 - D_{\phi_i}(o_i, a_i)\right)\right]. \tag{22}$$

Функция вознаграждения агента *i* основана на дискриминаторе:

$$R_i(o_i, a_i) = f_{\phi_i}(o_i, a_i) - \log\pi_{\theta_i}(a_i|\, o_i)\,. \tag{23}$$

Функция потерь для политики агента *i*:

$$L_G(\theta_i) = E_{\tau_i\sim\pi_{\theta_i}}\left[\sum_{t=0}^{T}\left(f_{\phi_i}(o_{i,t}, a_{i,t}) - \log\pi_{\theta_i}(a_{i,t}|\, o_{i,t})\right)\right]. \tag{24}$$

ЗАКЛЮЧЕНИЕ

Мультиагентный AIRL позволяет моделировать сложные и многоступенчатые атаки, где несколько агентов взаимодействуют друг с другом и с окружающей средой. Этот подход значительно расширяет возможности тестирования на проникновение, предоставляя более реалистичные сценарии и позволяя учитывать динамику атак, которая часто остается незамеченной при использовании традиционных методов. Благодаря возможности координации действий между агентами мультиагентные системы могут выявлять уязвимости, которые были бы недоступны для одного агента, действующего в изоляции.

Одной из ключевых особенностей мультиагентного AIRL является использование многоуровневого обучения, что позволяет делить сложные задачи на иерархические уровни и обеспечивать более эффективную координацию действий агентов. Это особенно важно в условиях тестирования на проникновение, где необходимо учитывать разнообразные аспекты системы – от сети до приложений и данных. Многоуровневое обучение также упрощает решение задач, связанных с моделированием атак на разных уровнях абстракции, что делает тестирование на проникновение более комплексным и точным.

Обучение с частичным наблюдением, также реализуемое в рамках мультиагентного AIRL, позволяет агентам принимать решения на основе неполной информации о среде. Это приближает процесс тестирования на проникновение к реальным условиям, в которых злоумышленники не всегда имеют полный доступ к данным о цели. Такой подход делает анализ более реалистичным и эффективным, поскольку агенты могут адаптироваться к неопределенности и принимать обоснованные решения даже при наличии ограниченной информации.

Таким образом, мультиагентный подход в AIRL представляет собой мощный и гибкий инструмент для тестирования на проникновение, который значительно расширяет возможности анализа и выявления уязвимостей в современных информационных системах. Применение этого подхода позволяет повысить точность, адаптивность и глубину анализа, что в конечном итоге способствует улучшению безопасности систем и снижению рисков кибератак. В условиях быстро меняющегося ландшафта угроз мультиагентный AIRL является перспективным направлением, которое может существенно изменить подходы к обеспечению информационной безопасности и стать основой для будущих исследований и разработок в области тестирования на проникновение.

СПИСОК ЛИТЕРАТУРЫ

1. *Fu J., Luo K., Levine S.* Learning robust rewards with adversarial inverse reinforcement learning // arXiv e-prints. – 2017. – arXiv: 1710.11248.

2. *Sychugov A., Grekov M.* Automated penetration testing based on adversarial inverse reinforcement learning // 2024 International Russian Smart Industry Conference (SmartIndustryCon). – IEEE, 2024. – P. 373–377.

3. *Yu L., Song J., Ermon S.* Multi-agent adversarial inverse reinforcement learning // Proceedings of Machine Learning Research. – 2019. – Vol. 97: Proceedings of the 36th International Conference on Machine Learning, Long Beach, California. – P. 7194–7201.

4. *Chen J., Lan T., Aggarwal V.* Hierarchical adversarial inverse reinforcement learning // IEEE Transactions on Neural Networks and Learning Systems. – 2023. – DOI: 10.1109/TNNLS.2023.3305983.

5. Multi-task hierarchical adversarial inverse reinforcement learning / J. Chen, D. Tamboli, T. Lan, V. Aggarwal // Proceedings of Machine Learning Research. – 2023. – Vol. 202: Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii. – P. 4895–4920.

6. Adversarial reinforcement learning under partial observability in autonomous computer network defence / Y. Han, D. Hubczenko, P. Montague, O. De Vel, T. Abraham, B.I.P. Rubinstein, C. Leckie, T. Alpcan, S. Erfani // 2020 International Joint Conference on Neural Networks (IJCNN). – IEEE, 2020. – P. 1–8. – DOI: 10.1109/IJCNN48605.2020.9206634.

7. *Choi J.D., Kim K.E.* Inverse reinforcement learning in partially observable environments // Journal of Machine Learning Research. – 2011. – Vol. 12 (1). – P. 691–730.

Греков Михаил Михайлович, ассистент кафедры информационной безопасности Тульского государственного университета. Основное направление научных исследований – применение машинного обучения и нейронных сетей в области информационной безопасности, тестирование на проникновение. E-mail: grekov.web@yandex.ru

Сычугов Алексей Алексеевич, заведующий кафедрой информационной безопасности Тульского государственного университета. Область научных интересов – методы и алгоритмы оперативного обнаружения опасных состояний промышленных объектов, информационная безопасность. E-mail: xru2003@list.ru

DOI: 10.17212/2782-2230-2024-3-21-33

Multiagent penetration testing based on AIRL*

M.M. Grekov[1], A.A. Sychugov[2]

[1] *Tula State University, 92 Lenina Avenue, Tula, 300012, Russian Federation, assistant at the Department of Information Security. E-mail: grekov.web@yandex.ru*
[2] *Tula State University, 92 Lenina Avenue, Tula, 300012, Russian Federation, Head of the Department of Information Security. E-mail: xru2003@list.ru*

This paper explores the application of a multi-agent approach based on the Adversarial Inverse Reinforcement Learning (AIRL) method for penetration testing in information systems. Theoretical aspects of multi-agent AIRL are discussed, including the ability to model complex, multi-stage attacks, coordinate agent actions, and learn with partial observability, which accounts for limitations in information access. The practical application of this approach will demonstrate its effectiveness in identifying vulnerabilities, providing a deeper and more accurate security analysis.

Keywords: Adversarial Inverse Reinforcement Learning, information security, penetration testing, automation, multi-agent learning, partial observation, machine learning, neural networks

REFERENCES

1. Fu J., Luo K., Levine S. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv e-prints*, 2017, arXiv: 1710.11248.

2. Sychugov A., Grekov M. Automated penetration testing based on adversarial inverse reinforcement learning. *2024 International Russian Smart Industry Conference (SmartIndustryCon)*. IEEE, 2024, pp. 373–377.

3. Yu L., Song J., Ermon S. Multi-agent adversarial inverse reinforcement learning. *Proceedings of Machine Learning Research*, 2019, vol. 97: *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, California, pp. 7194–7201.

4. Chen J., Lan T., Aggarwal V. Hierarchical adversarial inverse reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. DOI: 10.1109/TNNLS.2023.3305983.

5. Chen J., Tamboli D., Lan T., Aggarwal V. Multi-task hierarchical adversarial inverse reinforcement learning. *Proceedings of Machine Learning Research*, 2023, vol. 202: *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, Hawaii, pp. 4895–4920.

* *Received 10 August 2024.*

6. Han Y., Hubczenko D., Montague P., De Vel O., Abraham T., Rubinstein B.I.P., Leckie C., Alpcan T., Erfani S. Adversarial reinforcement learning under partial observability in autonomous computer network defense. *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9206634.

7. Choi J.D., Kim K.E. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 2011, vol. 12 (1), pp. 691–730.

Для цитирования:

Греков М.М., Сычугов А.А. Мультиагентное тестирование на проникновение на основе AIRL // Безопасность цифровых технологий. – 2024. – № 3 (114). – С. 21–33. – DOI: 10.17212/2782-2230-2024-3-21-33.

For citation:

Grekov M.M., Sychugov A.A. Mul'tiagentnoe testirovanie na proniknovenie na osnove AIRL [Multi-agent penetration testing based on AIRL]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 3 (114), pp. 21–33. DOI: 10.17212/2782-2230-2024-3-21-33.

БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2024. – № 3 (114). – 34–52

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.8 DOI: 10.17212/2782-2230-2024-3-34-52

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ПОСТРОЕНИЯ СИСТЕМ ОБРАБОТКИ ЗАШИФРОВАННЫХ ДАННЫХ И ИХ СРАВНЕНИЕ ДЛЯ РЕШЕНИЯ ЗАДАЧ МАШИННОГО ОБУЧЕНИЯ*

ЛАПИНА М.А.[1], АРДЕЕВ Д.Ю.[2], ЛАПИН В.Г.[3]

[1] *355017, РФ, г. Ставрополь, ул. Пушкина, 1, ФГАОУ ВО «Северо-Кавказский федеральный университет», доцент кафедры информационной безопасности автоматизированных систем. E-mail: mlapina@ncfu.ru*
[2] *355017, РФ, г. Ставрополь, ул. Пушкина, 1, ФГАОУ ВО «Северо-Кавказский федеральный университет», ассистент кафедры информационная безопасность автоматизированных систем. E-mail: ardeev.dima345@gmail.com*
[3] *355017, РФ, г. Ставрополь, ул. Пушкина, 1, ФГАОУ ВО «Северо-Кавказский федеральный университет», кандидат физико-математических наук, доцент кафедры вычислительной математики и кибернетики. E-mail: vitlx@yandex.ru*

В статье рассмотрены два метода построения систем обработки зашифрованных данных, основанные на гомоморфном шифровании и разделительных вычислениях. Каждый метод отличается представленным алгоритмом и моделью обработки данных. Рассмотрены базовые операции над зашифрованными данными, а именно умножение и сложение. Для исследования использовались следующие библиотеки: TenSEAL, TensorFlow и PyTorch. Библиотека TenSEAL является инструментом полного гомоморфного шифрования, созданного для языка программирования C++, но адаптированного под использующийся в исследовании язык программирования Python. Эта библиотека позволяет использовать метод полного гомоморфного шифрования в построении вычислительной модели, задачей которой будет обработка зашифрованных данных. Для реализации метода разделительных вычислений, более известного как multy-party computation, будет использоваться библиотека TensorFlow, позволяющая создавать несколько тензоров и одновременно обучать их, что, в свою очередь, дает возможность реализовать принцип разделительных вычислений.

Ключевые слова: шифрование, конфиденциальность, гомоморфное шифрование, разделительные вычисления, тензоры, шифротексты

* Статья получена 13 августа 2024 г.

ВВЕДЕНИЕ

В современном мире информационные технологии стали неотъемлемой частью человеческой жизни. Причиной этого является технологический скачок в области компьютерных технологий, а также распространение широкополосных информационных систем, таких как интернет. Этот процесс способствует широкой автоматизации объектов социальной и критической инфраструктуры современного государства, а также связан с увеличением объема потока данных, необходимого для их корректной работы. Из этого следует, что информация – крайне важный ресурс, требующий обеспечения безопасности и конфиденциальности данных.

В настоящее временя актуальны два вида методов построения систем обработки зашифрованных данных – это гомоморфное шифрование и разделительные вычисления. Построение моделей систем для работы с зашифрованными данными проведем с использованием соответствующих библиотек языка программирования Python.

1. ОПИСАНИЕ РАБОТЫ

1.1. МЕТОДЫ ОБРАБОТКИ ЗАШИФРОВАННЫХ ДАННЫХ

Гомоморфное шифрование [3] – это метод шифрования данных, позволяющий проводить различные математические операции, не выполняя дешифрования информации, тем самым сохраняя конфиденциальность сведений. Для верного выполнения гомоморфного шифрования данных должны быть выполнены два условия: условие корректности и условие конфиденциальности. Условие корректности – это условие, при котором результат математических вычислений над исходными данными возможно получить, если будет проведена замена математической операции на последовательный алгоритм, проводящий вычислительные операции над зашифрованными или расшифрованными данными. Условие конфиденциальности – это условие, при котором операция, проводимая над данными, и промежуточные результаты не должны быть расшифрованы и не должны предоставлять дополнительные сведения, которые могли бы способствовать преждевременному дешифрованию информации, а следовательно, и потери данных. Этот метод позволяет проводить над зашифрованными переменными как алгебраические операции, так и булевы. Исходя из этого имеются следующие типы гомоморфного шифрования.

Частичное гомоморфное шифрование (англ. Somewhat homomorphic encryption, SHE) – метод, поддерживающий операции одного конкретного типа над зашифрованными данными.

Полностью гомоморфное шифрование (англ. Full Homomorphic Encryption, FHE) – метод, поддерживающий операции различных типов над зашифрованными данными.

Для построения систем обработки зашифрованных данных будет использоваться метод полного гомоморфного шифрования FHE. Он имеет ряд определенных достоинств:

- гибкость вычислений – поддержка выполнения любых вычислительных операций над зашифрованными данными;
- минимизация потери конфиденциальности данных. Поскольку все операции происходят в зашифрованном виде, открытость вычислительных узлов минимизируется. Даже при утечке данные остаются скрытыми от мошенников из-за шифрования;
- обеспечение приватности в облачных вычислениях. Подходит для выполнения вычислений на удаленных серверах либо в закрытых компьютерных сетях, минимизирует риски потери данных.

Недостатки данного метода:

- высокие вычислительные затраты. Операции умножения и сложения зашифрованных элементов зачастую проходят намного дольше в сравнении с незашифрованными. Также на производительность влияет непосредственно объем данных (чем он больше, тем медленнее проходит операция);
- большие размеры шифротекстов. Шифрованные тексты, созданные с применением гомоморфного шифрования, по размеру превышают исходные данные;
- ограниченное количество операций. Несмотря на то что полное гомоморфное шифрование FHE поддерживает вычисления при помощи любых математических операций, подавляющее большинство задач решается ограниченным числом операций (например, сложением). Это свойство накладывает ограничения для реализации многоуровневых вычислительных задач, требующих разделения на отдельные подоперации.

Эти недостатки делают метод гомоморфного шифрования сложным для широкого практического применения в области вычислений над зашифрованными данными.

Безопасные разделительные вычисления (англ. Secure Multy-Party Computation, MPC) – это криптографический метод, обеспечивающий возможность выполнения вычислений одновременно несколькими операторами без необходимости раскрытия зашифрованной информации друг другу. Каждая сторона имеет фрагменты сокрытых данных, при объединении которых можно получить результат без раскрытия конфиденциальной информации.

Главными аспектами МРС являются следующие понятия:

- конфиденциальность. Оператор не имеет никакой информации о входных данных других операторов, кроме той, которую он подает на ввод и может получить при завершении вычислений;
- корректность. Результат вычислительных операций должен быть корректным по отношению к тому результату, который возможно было бы получить без использования метода разделительных вычислений;
- надежность. Отказоустойчивость системы и способность купировать попытки перехвата данных злоумышленниками. Она должна обеспечить корректность вычислений и верность конечного результата, несмотря на возникшие проблемы;
- согласованность. Операторы одновременно получают результат для предотвращения несанкционированного использования конечного результата злоумышленником либо же недобросовестным оператором для исключения возможности утечки информации;
- анонимность. Сокрытие информации об операторах, участвующих в предоставлении данных для разделительных вычислений.

Перечисленные понятия можно отнести к преимуществам данного метода, однако у него также имеется и ряд недостатков, с которыми сталкиваются большинство разработчиков, применяющих разделительные вычисления в своих проектах.

Перечень основных недостатков:

- зависимость от операторов. Надежность МРС зависит от поведения операторов, участвующих в вычислениях. При любых отклонениях от нормы ресурсоемкость вычислений повышается, также сбой у одного из участников вычислений может нарушить весь процесс;
- ограниченная масштабируемость. Методы МРС трудномасштабируемы для большого количества операторов, участвующих в операциях, что увеличивает время работы программного кода, а также ресурсоемкость, негативно сказывается на скорости вычислений, а также повышает риски выхода из строя алгоритма, так как возрастает возможность системной неисправности, способной нарушить работоспособность алгоритма.

Несмотря на потенциал разделительных вычислений в области обеспечения безопасности и конфиденциальности вычислений, проводимых над зашифрованными данными, для реализации этого метода на практике необходимо учитывать всевозможные угрозы и ограничения, которые могут способствовать нарушению работоспособности системы, построенной при использовании метода.

1.2. ПЕРЕЧЕНЬ БИБЛИОТЕК

Для создания моделей машинного обучения, на основе которых будут применяться методы, используются две наиболее распространенные библиотеки – TensorFlow и PyTorch. Остальные библиотеки используют поверх указанных библиотек, тем самым обеспечивая оптимизацию ресурсов системы, на которой запускается модель. Рассмотрим каждую библиотеку в отдельности.

TensorFlow – это открытая библиотека, созданная для машинного обучения и адаптированная компанией Google. Она обеспечивает широкий функционал модели машинного обучения. Этот метод для проведения расчетов использует структуру статических графов, узлы которых представляют собой математические операции, а ребра – это многомерные массивы (тензоры). Такой метод позволяет проводить эффективные вычисления между устройствами электронной вычислительной машины, такими как центральный процессор и графический процессор. Также немаловажным отличием является ее гибкость. TensorFlow поддерживает различные уровни абстракции: от низкоуровневых API для сложных математических задач и до высокоуровневых API, таких как Keras, которые упрощают создание и обучение моделей.

Главным преимуществом библиотеки TensorFlow для такого исследования является его дополнительный функционал, предназначенный для работы с информационными моделями, работающими методами разделительных вычислений.

PyTorch – это библиотека, созданная для машинного обучения компанией Facebook и имеющая открытый код. Для вычислений использует динамический граф, позволяющий изменять структуру данных при выполнении обучения модели – это делает библиотеку удобной для работы с различными типами данных и различными объемами, однако это может сказаться негативно на качестве получаемых вычислений. Из преимуществ можно назвать открытый доступ к уже готовым и обученным моделям нейронных сетей, что позволяет сразу приступать к различным тестам и экспериментам над ними.

Из обеих представленных библиотек для построения моделей вычислительных систем наиболее подходящей для текущего исследования является TensorFlow. Эта библиотека предоставляет расширенный функционал создания информационной модели, а также имеет богатую экосистему для развертывания моделей.

TenSEAL – это библиотека, изначально написанная на языке C++, позже была адаптирована под развертку языка Python. Она специализируется на реализации алгоритмов гомоморфного шифрования. Главной особенностью биб-

лиотеки является то, что она работает с открытой библиотекой TensorFlow. Основные преимущества библиотеки:

- реализация функций гомоморфного шифрования на более быстром языке программирования, так как язык Python не является строго типизированным языком программирования;
- производительность. Библиотека TenSEAL имеет высокое быстродействие при работе над зашифрованными данными, поскольку основная часть функционала написана на языке C++, имеющем строгую типизацию данных и тем самым обеспечивающим быстрые вычисления;
- масштабируемость. Библиотека TenSEAL предназначена для работы с большими объемами данных, что позволяет использовать ее в областях, сопряженных с выполнением многоуровневых задач.

2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

2.1. РЕАЛИЗАЦИЯ МЕТОДОВ

Алгоритм работы с инструментами, необходимыми для проведения вычислительных операций над зашифрованными данными при помощи методов гомоморфного шифрования и разделительных вычислений, представлен ниже.

1. Для установки библиотек TenSEAL и TensorFlow необходимо запустить среду разработки для языка программирования Python.

2. В консоли среды разработки прописать команду **pip install** и указать названия библиотек.

3. После установки библиотеки импортировать модули библиотек непосредственно в код при помощи команды **from** (название библиотеки) **import** (название модулей).

После выполнения предыдущих шагов появляется возможность начать исследование операций над зашифрованными данными.

Для выполнения построения модели нейронной сети, использующей метод разделительных вычислений, как было описано выше, будем использовать библиотеку TensorFlow [1]. Эта библиотека является очень удобным инструментом и имеет самый широкий перечень функций в сравнении с аналогами.

В качестве операционной системы, применяемой в исследовании нейронной сети, была выбрана Windows 10, среда разработки – google colabrory [12],

язык программирования – Python [13], для обучения модели будет использоваться датасет MNIST на 1000 изображений.

Ниже приведен алгоритм построения системы, основанной на разделительных вычислениях, для работы с зашифрованными данными.

1. Производится установка основных библиотек, необходимых для построения модели вычислительной системы, через команду **pip** в области программирования.

2. Далее импортируем в код модули библиотек при помощи команды **import**.

3. Для обучения созданной модели необходимо загрузить данные из датасета MNIST, после чего полученные переменные преобразуются путем деления на 255, тем самым придут к диапазону от нуля до единицы. Эта операция продемонстрирована на рис. 1.

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)
```

Рис. 1. Преобразование данных

Как показано на рис. 2, данные разделяются на две подвыборки при помощи функции **train_test_split**, полученной из библиотеки **sklearn**.

```
x_train_1, x_train_2, y_train_1, y_train_2 = train_test_split(x_train, y_train, test_size=0.5, random_state=42)
x_test_1, x_test_2, y_test_1, y_test_2 = train_test_split(x_test, y_test, test_size=0.5, random_state=42)
```

Рис. 2. Разделение данных

Выполняется формирование базовой модели с помощью функции **create_model** и компиляция ее с использованием **tensorflow.keras.Sequential**, после чего компилируется с использованием оптимизатора Adam и функции потерь **categorical_crossentropy** (рис. 3).

На рис. 4 показано, как создается первая модель с помощью функции **create_model()** и обучается при помощи валидации **validation_data(x_test_1, y_test_1)**.

```
def create_model():
    model = Sequential([
        Flatten(input_shape=(28, 28)),
        Dense(512, activation='relu'),
        Dropout(0.2),
        Dense(10, activation='softmax')
    ])
    model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
```

Рис. 3. Формирование базовой модели

```
model_1 = create_model()
history_1 = model_1.fit(x_train_1, y_train_1, epochs=5, validation_data=(x_test_1, y_test_1))
```

Рис. 4. Создание первой модели

Затем создается вторая модель по тому же принципу, что и предыдущая (рис. 5).

При помощи функции **ensemble_predictions** происходит объединение примерной точности обеих моделей.

```
def ensemble_predictions(models, data):
    predictions = [model.predict(data) for model in models]
    averaged_predictions = np.mean(predictions, axis=0)
    return np.argmax(averaged_predictions, axis=1)
```

Рис. 5. Объединение примерной точности моделей

После проведения всех вышеперечисленных операций производится создание списка моделей (рис. 6), куда добавляются предыдущие модели.

```
models = [model_1, model_2]
```

Рис. 6. Список моделей

Оценивается точность операции шифрования и расшифровывания на тестовом наборе данных.

Выполняется вывод данных формата: точность, скорость работы кода, затраты оперативной памяти для проведения вычислений методом разделительных вычислений.

В табл. 1 представлен полученный результат.

Таблица 1

Полученные данные в результате исследования модели MPC

Точность в зашифрованной модели	0.9792
Время работы	174.1779 с
Использовано ОЗУ	1863.18 Мб

Модель показывает высокую скорость работы алгоритма шифрования, построенного по принципу MPC. За счет распределения итоговой вычислительной нагрузки на две подмодели основной алгоритм достигает повышенной точности при минимально затраченном времени. Из этого можно сделать вывод, что метод разделительных вычислений – один из самых мощных вычислительный инструментов в области булевых и алгебраических операций над зашифрованными данными.

Исследование метода гомоморфного шифрования будет основываться на запатентованной готовой модели. Для реализации поставленной задачи будет изменен объем данных, поступающих на вход, что позволит выполнить более точный анализ.

Для выполнения построения модели, основанной на методе гомоморфного шифрования, будет использован тот же язык программирования, что и для прошлой модели, – Python. Среда программирования – google colab, операционная система персонального компьютера – Windows 10, для чистоты эксперимента в данной модели также будет использоваться датасет MNIST на 1000 изображений.

В ходе проведенного анализа методов гомоморфного шифрования для исследования был выбран метод полного гомоморфного шифрования (англ. Full Homomorphic Encryption, FHE). Наиболее подходящая библиотека, реализующая принцип FHE, – это библиотека TenSEAL, так как она изначально создавалась на языке программирования C++, но имеет развертку для языка Python, что существенно облегчает ее применение в текущем исследовании.

Ниже представлен алгоритм построения системы.

Производится установка основных библиотек, необходимых для построения модели вычислительной системы через команду **pip** в среде программирования.

Далее импортируем в код модули библиотек при помощи команды **import**.

Загрузка тренировочных и тестовых данных через команду **datasets.MNIST** и их преобразование в тензоры при помощи команды **transforms.ToTensor()** показана на рис. 7.

```
# Загрузка и подготовка данных MNIST
train_data = datasets.MNIST('data', train=True, download=True, transform=transforms.ToTensor())
test_data = datasets.MNIST('data', train=False, download=True, transform=transforms.ToTensor())
```

Рис. 7. Загрузка тренировочного и тестового датасета

Создание загрузчика данных, необходимого для тренировочного набора данных, выполняется через команду **train_loader** с размером блока 64, а подвыборка набора тестовых данных – через команду **test_loader** с размером блока в 10 элементов. Участок кода продемонстрирован на рис. 8.

```
# Создание загрузчиков данных
train_loader = torch.utils.data.DataLoader(train_data, batch_size=batch_size, shuffle=True)
K = 10  # Размер тестовой выборки
subsample_test_indices = torch.randperm(len(test_data))[:K]
test_loader = torch.utils.data.DataLoader(test_data, batch_size=batch_size, sampler=torch.u
```

Рис. 8. Создание загрузчика

Создание вычислительной модели и ее определение через команду **class ConvNet**, а также ее инициализация представлены на рис. 9.

```
# Определение модели нейронной сети
class ConvNet(torch.nn.Module):
    def __init__(self, hidden=64, output=10):
        super(ConvNet, self).__init__()
        self.conv1 = torch.nn.Conv2d(1, 4, kernel_size=7, padding=0, stride=3)
        self.fc1 = torch.nn.Linear(256, hidden)
        self.fc2 = torch.nn.Linear(hidden, output)
```

Рис. 9. Инициализация модели

Обучение модели благодаря функции **Train** для будущего выполнения алгебраических операций над зашифрованными тензорами показано на рис. 10.

```
# Функция для обучения модели
def train(model, train_loader, criterion, optimizer, n_epochs=10):
    model.train()
    for epoch in range(1, n_epochs + 1):
        train_loss = 0.0
        for data, target in train_loader:
            optimizer.zero_grad()
            output = model(data)
            loss = criterion(output, target)
            loss.backward()
            optimizer.step()
            train_loss += loss.item()
        train_loss = train_loss / len(train_loader)
        print('Epoch: {} \tTraining Loss: {:.6f}'.format(epoch, train_loss))
    model.eval()
    return model
```

Рис. 10. Функция обучения модели

Функция тестирования модели для проверки работоспособности, а также вычисление средней точности показаны на рис. 11.

```
# Функция для тестирования модели
def test(model, test_loader, criterion):
    test_loss = 0.0
    class_correct = list(0. for i in range(10))
    class_total = list(0. for i in range(10))
    model.eval()
    for data, target in test_loader:
        output = model(data)
        loss = criterion(output, target)
        test_loss += loss.item()
        _, pred = torch.max(output, 1)
        correct = np.squeeze(pred.eq(target.data.view_as(pred)))
        for i in range(len(target)):
            label = target.data[i]
            class_correct[label] += correct[i].item()
            class_total[label] += 1
    test_loss = test_loss / len(test_loader)
    print(f'Test Loss: {test_loss:.6f}\n')
    for label in range(10):
        print(f'Test Accuracy of {label}: {int(100 * class_correct[label] / class_total[label])}% ({int(np.sum(class_correct[label]))}/{int(np
    print(f'\nTest Accuracy (Overall): {int(100 * np.sum(class_correct) / np.sum(class_total))}% ({int(np.sum(class_correct))}/{int(np.sum(cla
```

Рис. 11. Функция тестирования модели

Создание зашифрованной модели, содержащей преобразованные в шифротекст тензоры через команду **EncConvNet** и копирование весов из обученной модели Conv, показаны на рис. 12.

```
# Класс для зашифрованной нейронной сети
class EncConvNet:
    def __init__(self, torch_nn):
        # Копирование весов и смещений из обученной модели ConvNet
        self.conv1_weight = torch_nn.conv1.weight.data.view(
            torch_nn.conv1.out_channels, torch_nn.conv1.kernel_size[0],
            torch_nn.conv1.kernel_size[1]
        ).tolist()
        self.conv1_bias = torch_nn.conv1.bias.data.tolist()

        self.fc1_weight = torch_nn.fc1.weight.T.data.tolist()
        self.fc1_bias = torch_nn.fc1.bias.data.tolist()

        self.fc2_weight = torch_nn.fc2.weight.T.data.tolist()
        self.fc2_bias = torch_nn.fc2.bias.data.tolist()
```

Рис. 12. Создание зашифрованной модели

Эта модель имеет большее количество функций, необходимых для верного проведения шифрования данных и последующих операций над ними. Далее представлена таблица результатов работы системы обработки зашифрованных данных (табл. 2).

Таблица 2

Результат работы модели, основанной на методе гомоморфного шифрования

Точность в зашифрованной модели	0.9894
Время работы	650.1342 с
Использовано ОЗУ	2363.18 Мб

Как показывает практическое исследование модели, основанной на методе гомоморфного шифрования, быстродействие падает из-за специфики принципа работы. Данные поступают на вход единым пакетом, а не частями, поэтому сначала создается тензор, состоящий полностью из незашифрованной информации, а затем он преобразуется в зашифрованный блок, что создает дополнительную нагрузку на алгоритм и требует больше оперативной

памяти. Однако стоит отметить, что, несмотря на ресурсоемкость операции, конечный показатель точности довольно высок.

2.2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ

Оба метода позиционируются как инструменты для работы с зашифрованной информацией, а также предназначены для сохранения конфиденциальности данных. Наглядно сравним показатели быстродействия, ресурсоемкости и точности при помощи табл. 3.

Таблица 3

Сравнение моделей

Название метода	Точность	Быстродействие	Ресурсоемкость
МРС	0.9792	174.1779 с	1863.18 Мб
ГШ	0.9994	550.1342 с	2363.25 Мб
Разница	0.202	375,9563 с	500.07 Мб

Из вышеприведенных данных можем утверждать следующее: скорость быстродействия выше у модели, построенной на разделительных вычислениях, чем у модели, построенной на гомоморфном шифровании, примерно на 376 секунд при обработке датасета на 1000 изображений. Достигается это за счет того, что перед обработкой полученных данных методом разделительных вычислений сет элементов делится надвое и одновременно шифруется, в то время как алгоритм, построенный на гомоморфном шифровании, обрабатывает данные целиком, превращает полученный датасет в тензор, а уже потом начинает процесс шифрования. Также по этой причине алгоритму на МРС требуется меньше оперативной памяти, чем алгоритму на FHE. В то же время показатель точности у разделительных вычислений немного ниже, чем у FHE. Вызвано это потерями «крупиц» при раздельном шифровании датасета и разделительных вычислений. Для малых объемов информации потеря одного-трех процентов данных не столь критична, однако при работе с большими данными это может негативно сказаться на точности измерений.

В статье проведено исследование моделей нейронных сетей, построенных на основе методов гомоморфного шифрования, а именно метода полного гомоморфного шифрования и метода разделительных вычислений. Рассмот-

рены две модели: первая модель использует метод MPC, а вторая – метод FHE.

Анализ показывает, что каждая из моделей имеет свои преимущества и недостатки. Модель на основе MPC демонстрирует высокую скорость работы и меньшую ресурсоемкость благодаря параллельной обработке данных. Этот метод лучше подходит для задач, требующих быстрого выполнения и работы с небольшими объемами данных. Однако точность вычислений у этой модели несколько ниже из-за возможных потерь данных при параллельной обработке зашифрованных данных.

С другой стороны, модель на основе FHE отличается высокой точностью вычислений, что делает ее предпочтительной для задач, требующих обработки большого количества данных и не имеющих приоритета по времени выполнения. Однако этот метод требует значительных вычислительных ресурсов для обработки данных. Связано это с тем, что при использовании этого метода датасет поступает целиком, что требует большего времени для обработки данных на каждом этапе вычислений.

ЗАКЛЮЧЕНИЕ

В результате сравнения двух моделей установлено, что ни один из использующихся в исследовании методов не является универсальным. Выбор метода зависит от конкретных задач и требований к системе. Если приоритетом является быстродействие и обработка небольших объемов данных, лучше использовать метод MPC. Если же важна точность результатов, предпочтительнее использовать метод FHE, несмотря на его высокую ресурсоемкость.

Таким образом, в ходе исследования продемонстрировано, что оба метода имеют свои области применения и могут эффективно использоваться для решения различных задач машинного обучения с обработкой зашифрованных данных.

Подводя итоги, можно сказать, что с современным уровнем цифровизации и информатизации всех отраслей жизни человека всё более насущным становится вопрос сохранения конфиденциальности данных и работы с ними в зашифрованном виде. Основными методами для таких операций являются разделительные вычисления и гомоморфное шифрование, так как обеспечивают безопасность личной информации человека, но имеют ряд особенностей, которые порождают вопросы о том, какой метод эффективнее.

Относительно недавно человечество открыло для себя нейронные технологии, которые способны облегчить ряд задач, встающих перед ним.

Главным отличием таких технологий является то, что они не программируются в привычном понимании, а обучаются. Стоит отметить, что любое обучение – это работа с данными. Задача методов гомоморфного шифрования и разделительных вычислений состоит в том, чтобы сохранить безопасность и конфиденциальность информации, которая будет использоваться для машинного обучения.

По завершении исследования можно сделать вывод, что нет универсального метода работы с зашифрованными данными. Опираясь на полученные результаты, можно сказать, что разделительные вычисления больше подходят для решения задач, требующих быстродействия и работы с небольшим объемом данных. Гомоморфное шифрование больше подходит для получения не столько быстрого, сколько более точного результата. Обработка больших данных – трудоемкая операция с высокой точностью результатов.

СПИСОК ЛИТЕРАТУРЫ

1. TenSeal – Python FHE Library: сайт. – URL: https://pypi.org/project/tenseal/ (дата обращения: 28.08.2024).

2. Tensorflow – Python MPC Library: сайт. – URL: https://www.tensorflow.org/?hl=ru (дата обращения: 28.08.2024).

3. Полностью гомоморфное шифрование (обзор) / Л.К. Бабенко, Ф.Б. Буртыка, О.Б. Макаревич, А.В. Трепачева // Вопросы защиты информации. – 2015. – № 3 (110). – С. 3–26. – URL: https://elibrary.ru/item.asp?id=24833959 (дата обращения: 28.08.2024).

4. What is Secure Multiparty Computation // Inpher. Technology. – URL: https://inpher.io/technology/what-is-secure-multiparty-computation/ (accessed: 28.08.2024).

5. Multy-Party Computation method / GitHub. – URL: https://github.com/rdragos/awesome-mpc (accessed: 28.08.2024).

6. Multy-Party Computation encrypted / GitHub. – URL: https://github.com/tf-encrypted/tf-encrypted (accessed: 28.08.2024).

7. Full homomorphic encryption in python / GitHub. – URL: https://github.com/ViktorAxelsen/TFE-GNN (accessed: 28.08.2024).

8. PyTorch framework: website. – URL: https://pytorch.org/ecosystem/ (accessed: 28.08.2024).

9. *Nitin Kendre*. 7 Essential techniques for data preprocessing using python: a guide for data scientists / DEV Community. – URL: https://dev.to/newbie_coder/

7-essential-techniques-for-data-preprocessing-using-python-a-guide-for-data-scientists-3ijk (accessed: 28.08.2024).

10. AI Confidential: How can machine learning on encrypted data improve privacy protection? / Ericsson blog. – URL: https://www.ericsson.com/en/blog/2021/9/machine-learning-on-encrypted-data (accessed: 28.08.2024).

11. *Monteiro Tiago Capelo*. How homomorphic encryption works / FreeCodeCamp. – URL: https://www.freecodecamp.org/news/homomorphic-encryption-in-plain-english/ (accessed: 28.08.2024).

12. Google Colaboratory: website. – URL: https://colab.google/ (accessed: 29.08.2024).

13. Python 3.12.4 // Python: website. – URL: https://www.python.org/downloads/ (accessed: 29.08.2024).

14. *Crockett E.* Building machine learning models with encrypted data / Amazon Science Blog. – 2021. – January 5. – URL: https://www.amazon.science/blog/building-machine-learning-models-with-encrypted-data (accessed: 29.08.2024).

15. Federated Learning / Papers with Code: website. – URL: https://paperswithcode.com/task/federated-learning (accessed: 29.08.2024).

16. Dataset MNIST / Kaggel: website. – URL: https://www.kaggle.com/datasets/hojjatk/mnist-dataset (accessed: 29.08.2024).

17. Encrypted key-value database using homomorphic encryption / ZAMA. – URL: https://www.zama.ai/post/encrypted-key-value-database-using-homomorphic-encryption (accessed: 29.08.2024).

18. Программный комплекс для моделирования сверточных нейронных сетей, сохраняющих конфиденциальность, в условиях ограниченных вычислительных ресурсов: свидетельство о гос. регистрации программы для ЭВМ № 2024614441 / Лапина М.А., Фисенко Н.С., Бабенко М.Г. – URL: https://www1.fips.ru/ofpstorage/Doc/PrEVM/RUNWPR/000/002/024/614/441/2024614441-00001/document.pdf (дата обращения: 29.08.2024).

Лапина Мария Анатольевна, кандидат физико-математических наук, доцент, доцент кафедры информационной безопасности автоматизированных систем Северо-Кавказского федерального университета. E-mail: mlapina@ncfu.ru

Ардеев Дмитрий Юрьевич, ассистент кафедры информационной безопасности автоматизированных систем Северо-Кавказского федерального университета. E-mail: ardeev.dima345@gmail.com

Лапин Виталий Геннадьевич, кандидат физико-математических наук, доцент кафедры вычислительной математики и кибернетики Северо-Кавказского федерального университета. E-mail: vitlx@yandex.ru

DOI: 10.17212/2782-2230-2024-3-34-52

Comparative analysis of methods for building encrypted data processing systems and their comparison for solving machine learning problems*

Lapina M.A.[1], Ardeev D.Yu.[2], Lapin V.G.[3]

[1] *Federal State Autonomous Educational Institution of Higher Education "North Caucasus Federal University", 1 Pushkina Street, Stavropol, 355017, Russian Federation, associate professor of the department of information security of automated systems. E-mail: mlapina@ncfu.ru*

[2] *Federal State Autonomous Educational Institution of Higher Education "North Caucasus Federal University", 1 Pushkina Street, Stavropol, 355017, Russian Federation, assistant of the department information security of automated systems. E-mail: ardeev.dima345@gmail.com.*

[3] *Federal State Autonomous Educational Institution of Higher Education "North Caucasus Federal University", 1 Pushkina Street, Stavropol, 355017, Russian Federation, associate professor of the department of computational mathematics and cybernetics. E-mail: vitlx@yandex.ru*

The article discusses two methods for constructing encrypted data processing systems based on homomorphic encryption and separation calculations. Each method differs from each other in the presented algorithm and data processing model. Basic operations on encrypted data are considered, namely, multiplication and addition. The following libraries were used for the study: tenSEAL, tensorflow and PyTorch. The tenSEAL library is a fork of the full homomorphic encryption tool created for the C++ programming language, but adapted for the Python programming language used in the study. This library allows you to use the method of full homomorphic encryption in constructing a computational model, the task of which will be to process encrypted data. To implement the method of separation calculations, better known as multi-party computation, the tensorflow library will be used, which allows you to create several tensors and train them simultaneously, which in turn makes it possible to implement the principle of separation calculations.

Keywords: encryption, confidentiality, homomorphic encryption, separation computing, tensors, ciphertexts

* *Received 13 August 2024.*

REFERENCES

1. *TenSeal – Python FHE Library*. Website. Available at: https://pypi.org/project/tenseal/ (accessed 28.08.2024).

2. Tensorflow – Python MPC Library. Website. Available at: https://www.tensorflow.org/?hl=ru (accessed 28.08.2024).

3. Babenko L.K., Burtyka F.B., Makarevich O.B., Trepacheva A.V. Polnost'yu gomomorfnoe shifrovanie (obzor) [Fully homomorphic encryption (Review)]. *Voprosy zashchity informatsii = Information Security Questions*, 2015, no. 3 (110), pp. 3–26. Available at: https://elibrary.ru/item.asp?id=24833959 (accessed 28.08.2024).

4. Inpher. Technology. *What is Secure Multiparty Computation*. Available at: https://inpher.io/technology/what-is-secure-multiparty-computation/ (accessed 28.08.2024).

5. GitHub. *Multy-Party Computation method*. Available at: https://github.com/rdragos/awesome-mpc (accessed 28.08.2024).

6. GitHub. *Multy-Party Computation encrypted*. Available at: https://github.com/tf-encrypted/tf-encrypted (accessed 28.08.2024).

7. GitHub. *Full homomorphic encryption in python*. Available at: https://github.com/ViktorAxelsen/TFE-GNN (accessed 28.08.2024).

8. *PyTorch framework*. Website. Available at: https://pytorch.org/ecosystem/ (accessed 28.08.2024).

9. Nitin Kendre. *7 Essential techniques for data preprocessing using python: a guide for data scientists*. DEV Community. Available at: https://dev.to/newbie_coder/7-essential-techniques-for-data-preprocessing-using-python-a-guide-for-data-scientists-3ijk (accessed 28.08.2024).

10. Ericsson blog. *AI Confidential: How can machine learning on encrypted data improve privacy protection?* Available at: https://www.ericsson.com/en/blog/2021/9/machine-learning-on-encrypted-data (accessed 28.08.2024).

11. Monteiro Tiago Capelo. *How homomorphic encryption works*. FreeCodeCamp. Available at: https://www.freecodecamp.org/news/homomorphic-encryption-in-plain-english/ (accessed 28.08.2024).

12. *Google Colaboratory*. Website. Available at: https://colab.google/ (accessed 29.08.2024).

13. Python 3.12.4. Available at: https://www.python.org/downloads/ (accessed 29.08.2024).

14. Crockett E. Building machine learning models with encrypted data. *Amazon Science Blog*, 2021, January 5. Available at: https://www.amazon.science/blog/building-machine-learning-models-with-encrypted-data (accessed 29.08.2024).

15. Papers with Code. *Federated Learning*. Available at: https://papers-withcode.com/task/federated-learning (accessed 29.08.2024).

16. Kaggel. *Dataset MNIST*. Available at: https://www.kaggle.com/datasets/hojjatk/mnist-dataset (accessed 29.08.2024).

17. ZAMA. *Encrypted key-value database using homomorphic encryption*. Available at: https://www.zama.ai/post/encrypted-key-value-database-using-homomorphic-encryption (accessed 29.08.2024).

18. Lapina M.A., Fisenko N.S., Babenko M.G. *Programmnyi kompleks dlya modelirovaniya svertochnykh neironnykh setei, so-khranyayushchikh konfidentsial'nost', v usloviyakh ogranichennykh vychislitel'nykh* resursov [Software package for modeling privacy-preserving convolutional neural networks under limited computing conditions resources]. The Certificate on official registration of the computer program. No. 2024614441, 2024. Available at: https://www1.fips.ru/ofpstorage/Doc/PrEVM/RUNWPR/000/002/024/614/441/2024614441-00001/document.pdf (accessed 29.08.2024).

Для цитирования:

Лапина М.А., *Ардеев Д.Ю.*, *Лапин В.Г.* Сравнительный анализ методов построения систем обработки зашифрованных данных и их сравнение для решения задач машинного обучения // Безопасность цифровых технологий. – 2024. – № 3 (114). – С. 34–52. – DOI: 10.17212/2782-2230-2024-3-34-52.

For citation:

Lapina M.A., Ardeev D.Yu., Lapin V.G. Sravnitel'nyi analiz metodov postroeniya sistem obrabotki zashifrovannykh dannykh i ikh sravnenie dlya resheniya zadach mashinnogo obucheniya [Comparative analysis of methods for building encrypted data processing systems and their comparison for solving machine learning problems]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 3 (114), pp. 34–52. DOI: 10.17212/2782-2230-2024-3-34-52.

БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2024. – № 3 (114). – 53–62

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056 DOI: 10.17212/2782-2230-2024-3-53-62

К ВОПРОСУ РЕАЛИЗАЦИИ АЛГОРИТМОВ ПРОЕКТИРОВАНИЯ ЗАЩИЩЕННЫХ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ*

Г.В. ПОПКОВ

630102, РФ, г. Новосибирск, ул. Кирова, 86, Сибирский государственный университет телекоммуникаций и информатики, кандидат технических наук, заведующий кафедрой защиты информации в социальных системах. E-mail: glebpopkov@inbox.ru

Существующие подходы проектирования защищенных сетей передачи данных (ЗСПД) зачастую базируются на руководящих документах для проектирования сетей связи в широком смысле, в настоящий момент отсутствуют четкие регламенты для проектирования сетей связи в защищенном исполнении с точки зрения информационной безопасности. В процессе перспективного проектирования ЗСПД возникают проблемы поиска эффективных решений построения такого рода сетей связи и обеспечения информационной безопасности ЗСПД в части средств защиты информации (СЗИ) в заданных границах, определяемых измерениями информационной безопасности, таких как целостность, доступность, секретность информации, циркулирующей в ЗСПД. В статье предлагаются алгоритмы, базирующиеся на параметрах нестационарных гиперсетевых моделей, позволяющие реализовывать анализ и синтез ЗСПД, а также учитывать исходные данные по предполагаемым пользователям, сервисам / приложениям, реализуемым на ЗСПД в условиях внешних деструктивных воздействий (ВДВ).

Ключевые слова: сети передачи данных, информационная безопасность, целостность, доступность информации, гиперсетевые модели, устойчивость сетей передачи данных

ВВЕДЕНИЕ

Исходя из парадигмы проектирования защищенных сетей передачи данных (ЗСПД), устойчивых к внешним деструктивным воздействиям (ВДВ), в рамках систем принятия и поддержки решений (СППР) предлагаются алгоритмы, позволяющие использовать перспективные подходы для проектирования ЗСПД, работающих в условиях ВДВ.

* Статья получена 14 августа 2024 г.

1. ОПИСАНИЕ ЗАДАЧИ

Исходными данными при проектировании являются:

1) количество потребителей услуг ЗСПД;

2) набор и тип телекоммуникационных услуг, соответствующих потребителям;

3) территория, охватываемая ЗСПД (топооснова);

4) набор и типы систем связи, предполагаемых для создания ЗСПД;

5) типы нарушителей ИБ;

6) виды актуальных угроз на ЗСПД;

7) онтологии уязвимостей программно-аппаратного обеспечения предполагаемых систем связи;

8) риски, связанные с нарушением режима ИБ ЗСПД.

Для реализации анализа и синтеза ЗСПД автором была предложена нестационарная гиперсетевая [6, 7] модель *G-Net* в статье [1], используемая в системе проектирования (рис. 1), уровни модели задают параметры ЗСПД, соответствующие нижним четырем уровням модели *OSI* (open interconnect model) [2].

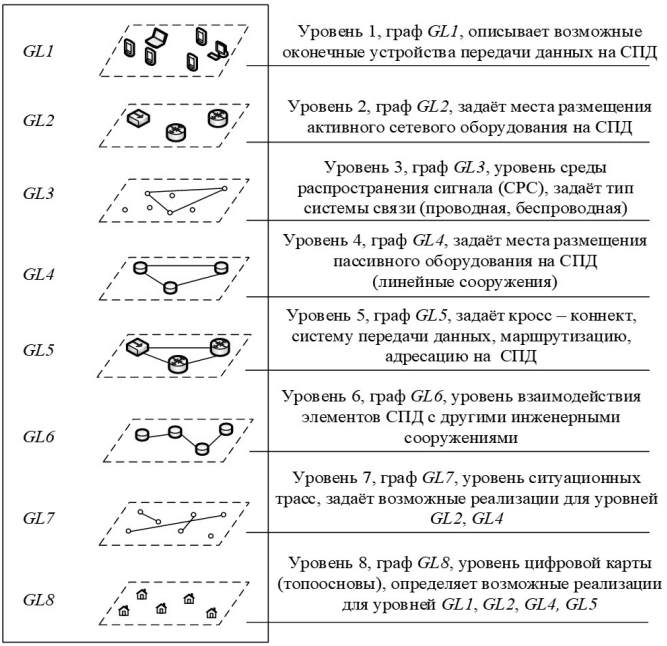

Рис. 1. Уровни нестационарной гиперсетевой модели *G-Net* СПД

2. АЛГОРИТМЫ D11, D12

На рис. 2 приведен укрупненный алгоритм D11, позволяющий проектировать защищенные СПД, устойчивые к ВДВ с ограничениями.

Шаг 1. Формирование структуры абонентов ЗСПД с учетом типа предполагаемых услуг $U_i[n..m]Gl_1$, теоретически количество пользователей ничем не ограничено либо ограничено параметрами уровня Gl_5.

Шаг 2. Определение границ предоставления услуг ЗСПД, ввод данных о местоположении пользователей на предполагаемой территории обслуживания сети ЗСПД вида $U_i[n..m]Gl_8$.

Шаг 3. Формирование структуры услуг ЗСПД, ввод данных о структуре и типе услуг на ЗСПД. Вывод промежуточных результатов по границам районов оказания услуг и необходимым потребностям в *i*-х услугах. Если результаты по услугам меньше требуемых, то переход на шаг 1.

Шаг 4. Формирование требований к заданному уровню измерений ИБ, целостности, доступности, структурной надежности, живучести элементов ЗСПД (программного обеспечения, коммутационных узлов, линейных сооружений) [8–10].

Шаг 5. Формирование профиля атаки с учетом модели нарушителя, модели угроз на сегменты ЗСПД с использованием баз *MITRE*, *CVSS*, ФСТЭК РФ.

Шаг 6. Задание узловой основы для транспортного сегмента ЗСПД с учетом уровней Gl_1, Gl_3, Gl_8.

Шаг 7. Решение о выборе системы связи с учетом вектора атаки на основании решений, принятых на уровнях Gl_2, Gl_4, Gl_5, Gl_6.

Шаг 8. Ранжирование критически значимых элементов ЗСПД согласно уровням модели *G-NET*, ввод данных о критически важных элементах *NE*.

Шаг 9. Формирование структуры проектируемой ЗСПД (кластера сети) с учетом выбранной системы связи и данных с уровней Gl_2, Gl_4, Gl_5, Gl_6.

Шаг 10. Решение задач маршрутизации на ЗСПД, ввод актуальных данных об известных угрозах на уровень маршрутизации (уровень Gl_5).

Шаг 11. Расчет значений устойчивости, проектируемой ЗСПД с учетом шага 7 (задание параметров надежности, живучести элементов ЗСПД).

Шаг 12. Определение показателей защищенности элементов ЗСПД с учетом выбранной системы связи, вектора атаки, применяемых методов избыточности (резервирования элементов ЗСПД), информации об уровнях Gl_2, Gl_4, Gl_5, Gl_6.

Шаг 13. Формирование структуры ЗСПД с учетом выбранной системы связи, вектора атаки, вывода результатов. Если результат удовлетворяет требованиям, то завершение работы алгоритма, если результат и условия не удовлетворяют шагу 4, то переход на шаг 5.

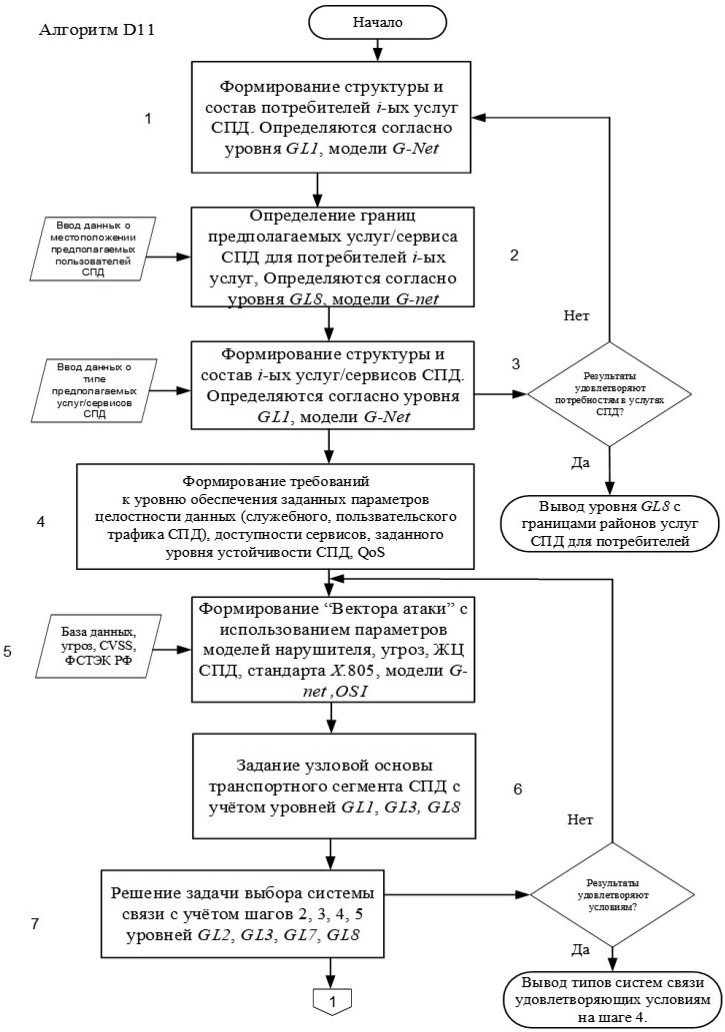

Рис. 2. Алгоритм D11

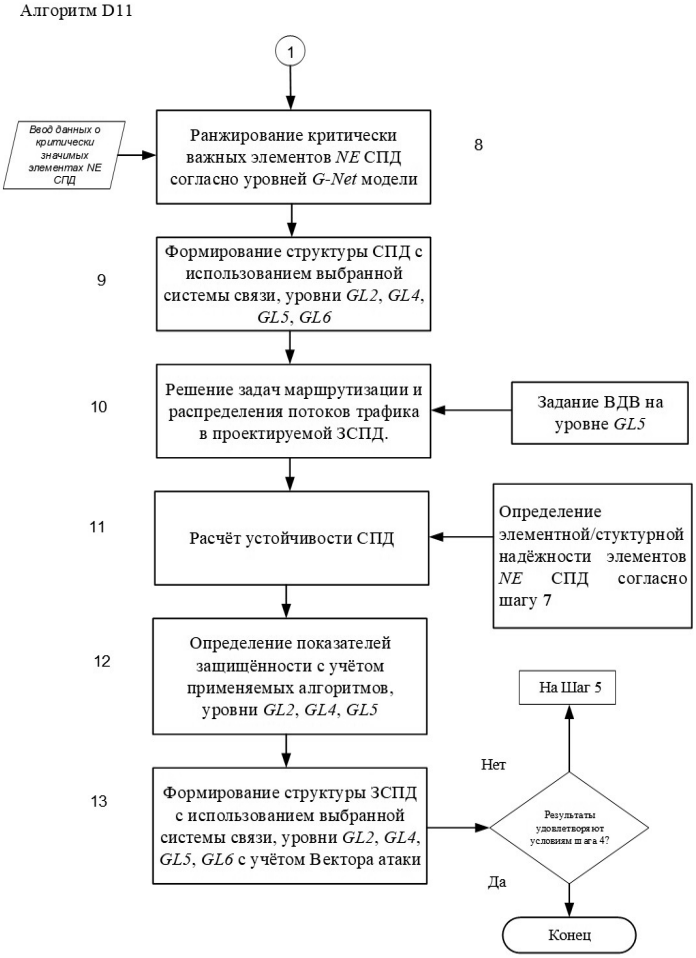

Рис. 2. Окончание

Рассмотренные алгоритмы являются частью системы СППР, основанной на задачном подходе [3, 4].

Очевидно, что предложенные алгоритмы являются частным случаем методики проектирования защищенных автоматизированных систем [5] и не противоречат нормативным документам, принятым в Российской Федерации.

Важнейшим этапом проектирования ЗСПД является выбор сертифицированных средств защиты информации, удовлетворяющих требованиям ФСТЭК Российской Федерации, по типу защищаемой информации. Выбор средств криптозащиты информации (СКЗИ) определяется на основании регулятивных документов ФСБ Российской Федерации.

На рис. 3 приведен укрупненный алгоритм D12 выбора эффективных средств защиты информации (СЗИ, СКЗИ), проектируемых ЗСПД в широком смысле.

Шаг 1. Анализ проектируемой ЗСПД по заданным параметрам (пропускная способность, система управления, система мониторинга инцидентов безопасности и т. д.).

Шаг 2. Определение критически важной информации, циркулирующей в ЗСПД.

Шаг 3. По итогам анализа определяется, требуется ли создание защищенной СПД.

Шаг 4. Определение класса защищенности.

Шаг 5. Выявление (определение) актуальных угроз для проектируемой ЗСПД.

Шаг 6. Разработка частных моделей нарушителя, частных моделей угроз.

Шаг 7. Формирование требований к средствам СЗИ, СКЗИ.

Шаг 8. Оценка возможности реализации требований по выбранным СЗИ.

Шаг 9. Анализ возможных уязвимостей сетевых элементов ЗСПД.

Шаг 10. Выбор сертифицированных средств СЗИ для проектируемой ЗСПД.

Шаг 11. Определяется, существует ли необходимый комплекс СЗИ. Если да, то переход на шаг 15; если нет, то переход на шаг 12.

Шаг 12. Формирование перечня сертифицированных СЗИ (СрСЗИ) для включения в систему защиты ЗСПД.

Шаг 13. Анализ эффективности выбранных средств СрСЗИ согласно выбранным критериям.

Шаг 14. Формирование / выбор оптимального состава СрСЗИ для комплекса мер защиты ЗСПД по заданным критериям.

Шаг 15. Формирование комплекса защитных средств с выбранными СрСЗИ.

Шаг 16. Анализ эффективности выбранных СрСЗИ.

Шаг 17. Формирование выбранных СрСЗИ для дальнейших проектов по планированию защищенных СПД.

Шаг 18. Определяется, удовлетворяют ли требованиям значения показателей СЗИ, СКЗИ. Если да, то переход на шаг 22; если нет, то переход на шаг 19.

Шаг 19. Оптимизация СЗИ, СКЗИ для проектируемой ЗСПД.

Шаг 20. Определяется, выполняются ли требования к СЗИ. Если да, то переход на шаг 21; если нет, то переход на шаг 10.

Шаг 21. Внедрение / тестирование СЗИ, СКЗИ на ЗСПД.

Шаг 22. Техническая эксплуатация комплексов СЗИ, СКЗИ ЗСПД.

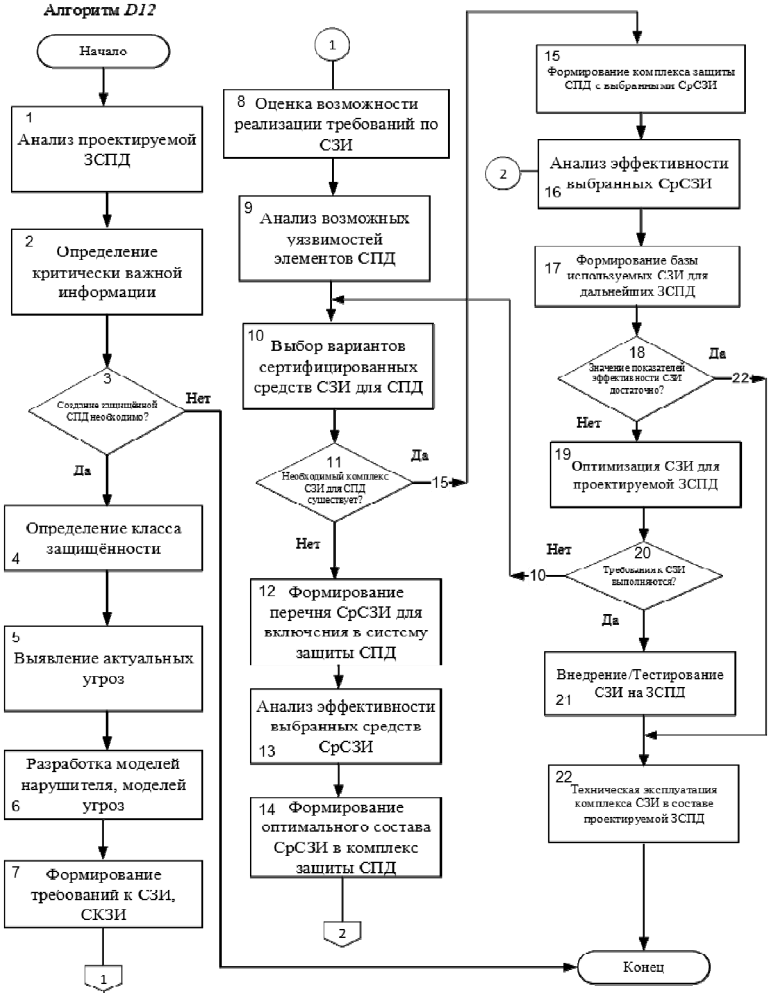

Рис. 3. Укрупненный алгоритм D12

ЗАКЛЮЧЕНИЕ

В статье рассмотрены два алгоритма проектирования защищенных сетей передачи данных, функционирующих в условиях внешних деструктивных воздействий. Эти алгоритмы являются составной частью системы принятия и поддержки решений, в конечном итоге определяющей состав, структуру, функционал защищенной сети связи, а также состав средств защиты информации, реализуемых на ЗСПД. В результате исследования был сделан вывод, что реализация нестационарных гиперсетевых моделей в части формирования задания ЗСПД позволяет проводить эффективный анализ и синтез ЗСПД в условиях ВДВ.

СПИСОК ЛИТЕРАТУРЫ

1. *Попков Г.В.* Перспективное проектирование сети абонентского доступа с использованием восьмиуровневой модели // Программные продукты и системы. – 2016. – № 2. – С. 139–145. – DOI: 10.15827/0236-235X.114.139-145.

2. Open System Interconnection model. CCITT X.200 (ITU-T X.200).

3. *Витяев Е.Е., Гончаров С.С., Свириденко Д.И.* О задачном подходе в искусственном интеллекте // Сибирский философский журнал. – 2019. – Т. 17, № 4. – С. 5–25. – DOI: 10.25205/2541-7517-2019-17-4-5-25.

4. *Витяев Е.Е., Гончаров С.С., Свириденко Д.И.* О задачном подходе в искусственном интеллекте и когнитивных науках // Сибирский философский журнал. – 2020. – Т. 18, № 2. – С. 5–29. – DOI: 10.25205/2541-7517-2020-18-2-5-29.

5. ГОСТ Р 51583–2014. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения. – М.: Стандартинформ, 2014.

6. *Попков В.К.* Математические модели связности. – Новосибирск: Изд-во ИВМиМГ СО РАН, 2006. – 490 с.

7. *Свами М., Тхуласираман К.* Графы, сети и алгоритмы. – М.: Мир, 1984. – 455 с.

8. *Назаров А.Н., Сычёв К.И.* Модели и методы расчета показателей качества функционирования узлового оборудования и структурно-сетевых параметров сетей связи следующего поколения. – Красноярск: Поликом, 2011. – 491 с.

9. ITU-T Recommendation X.800. Security Architecture for Open Systems Interconnection for CCITT Applications. – ITU, 1991.

10. ITU-T Recommendation X.805. Security Architecture for Systems providing end-to-end Communications. – ITU, 2003.

Попков Глеб Владимирович, кандидат технических наук, заведующий кафедрой защиты информации в социальных системах Сибирского государственного университета телекоммуникаций и информатики Область научных интересов – информационная безопасность. E-mail: glebpopkov@inbox.ru

DOI: 10.17212/2782-2230-2024-3-53-62

On the issue of implementing algorithms for designing secure transmission networks*

G.V. Popkov

Siberian state university of telecommunications and information science, 86 Kirova street, Novosibirsk, 630102, Russian Federation, head of the Department of Information Protection in Social Systems. E-mail: glebpopkov@inbox.ru

Existing approaches to the design of secure data transmission networks (DRN) are often based on guidelines for the design of communication networks in a broad sense, at the moment there are no clear regulations for the design of communication networks in a secure version from the point of view of information security. In the process of advanced design of the SPDS, problems arise in finding effective solutions for the construction of such communication networks and ensuring the information security of the SPDS in terms of information security tools (SIS) in given areas determined by information security measurements, such as the availability, accessibility, secrecy of information circulating in the SPDS. The article proposes algorithms based on the parameters of non-stationary hyper-network models, which make it possible to implement the analysis and synthesis of ZSPD, as well as take into account the initial data on the intended users, services/applications implemented on ZSPD under conditions of external destructive effects (VDV).

Keywords: data transmission networks, information security, integrity, information availability, hypernetwork models, stability of data transmission networks

REFERENCES

1. Popkov G.V. Perspektivnoe proektirovanie seti abonentskogo dostupa s ispol'zovaniem vos'miurovnevoi modeli [Advansed design of a customer access network using an 8-tier model]. *Programmnye produkty i sistemy = Software and Systems*, 2016, no. 2, pp. 139–145. DOI: 10.15827/0236-235X.114.139-145.

2. CCITT X.200. Standard (ITU-T X.200).*Open System Interconnection model*.

3. Vityaev E.E., Goncharov S.S., Sviridenko D.I. O zadachnom podkhode v iskusstvennom intellekte [On the task approach to artificial intelligence]. *Sibirskii*

* *Received 14 August 2024.*

filosofskii zhurnal = Siberian Journal of Philosophy, 2019, vol. 17 (4), pp. 5–25. DOI: 10.25205/2541-7517-2019-17-4-5-25.

4. Vityaev E.E., Goncharov S.S., Sviridenko D.I. O zadachnom podkhode v iskusstvennom intellekte i kognitivnykh naukakh [On the Task Approachin Artificial Intelligenceand Cognitive Sciences]. *Sibirskii filosofskii zhurnal = Siberian Journal of Philosophy*, 2020, vol. 18 (2), pp. 5–29. DOI: 10.25205/2541-7517-2020-18-2-5-29.

5. GOST R 51583–2014. *Zashchita informatsii. Poryadok sozdaniya avtomatizirovannykh sistem v zashchishchennom ispolnenii. Obshchie polozheniya* [State standard R 51583–2014. Information protection. Sequence of protected operational system formation. General]. Moscow, Standartinform Publ., 2014.

6. Popkov V.K. *Matematicheskie modeli svyaznosti* [Mathematical models of connectivity]. Novosibirsk, IVMiMG SO RAN Publ., 2006. 490 p.

7. Swamy M.N.S., Thulasiraman K. *Grafy, seti i algoritmy* [Graphs, networks and algorithms]. Moscow, Mir Publ., 1984. 455 p. (In Russian).

8. Nazarov A.N., Sychev K.I. *Modeli i metody rascheta pokazatelei kachestva funk-tsionirovaniya uzlovogo oborudovaniya i strukturno-setevykh parametrov setei svyazi sleduyushchego pokoleniya* [Models and methods for calculating performance indicators of nodal equipment and structural and network parameters of next-generation communication networks]. Krasnoyarsk, Polikom Publ., 2011. 491 p.

9. ITU-T Recommendation X.800. *Security Architecture for Open Systems Interconnection for CCITT Applications*. ITU, 1991.

10. ITU-T Recommendation X.805. *Security Architecture for Systems providing end-to-end Communications*. ITU, 2003.

Для цитирования:

Попков Г.В. К вопросу реализации алгоритмов проектирования защищённых сетей передачи данных // Безопасность цифровых технологий. – 2024. – № 3 (114). – С. 53–62. – DOI: 10.17212/2782-2230-2024-3-53-62.

For citation:

Popkov G.V. K voprosu realizatsii algoritmov proektirovaniya zashchishchennykh setei peredachi dannykh [On the issue of implementing algorithms for designing secure transmission networks]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 3 (114), pp. 53–62. DOI: 10.17212/2782-2230-2024-3-53-62.

БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2024. – № 3 (114). – 63–77

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056 DOI: 10.17212/2782-2230-2024-3-63-77

БПЛА КАК КИБЕРФИЗИЧЕСКАЯ СИСТЕМА*

Е.А. ВАСИЛЬЕВ[1], Е.С. АБРАМОВ[2]

[1] *347922, РФ, г. Таганрог, ул. Чехова, 2, Институт компьютерных технологий и информационной безопасности ИТА ЮФУ, аспирант кафедры безопасности информационных технологий. E-mail: evva@sfedu.ru*
[2] *347922, РФ, г. Таганрог, ул. Чехова, 2, Институт компьютерных технологий и информационной безопасности ИТА ЮФУ, заведующий кафедрой безопасности информационных технологий. E-mail: abramoves@sfedu.ru*

В настоящее время ученое сообщество дает нам отдельные определения беспилотных летательных аппаратов (БПЛА) и киберфизических систем. Эти определения полностью самодостаточны и не пересекаются между собой. Однако технический прогресс не стоит на месте, и узость данных определений не позволяет исследователям в полной мере применять некоторые определения из-за проблем в терминологии. Настоящая статья позволит преодолеть имеющиеся разногласия в терминологии и переносить свойства киберфизических систем на беспилотные летательные аппараты.

Ключевые слова: БПЛА, киберфизическая система, концептуальная модель, модели угроз, модели атак, модель kill-chain для КФС, уязвимости БПЛА, негативные последствия

ВВЕДЕНИЕ

В 2006 г. Хелен Джилл ввела термин для обозначения комплексов, состоящих из природных объектов, искусственных подсистем и контроллеров, – «киберфизические системы» [5].

Киберфизическая система (КФС) – это система, основанная на интеграции вычислений с физическими процессами. Встраиваемые компьютеры совместно с сетями осуществляют мониторинг и контроль за физическими процессами обычно путем передачи данных через узлы системы, где физические процессы влияют на вычисления, и наоборот. Это означает, что КФС может быть

* Статья получена 18 августа 2024 г.

определена как система, которая выполняет функции сбора, хранения, анализа, обработки и предоставления данных от устройств, взаимодействующих с физическими процессами и объектами, а также их тесную интеграцию с информационными технологиями в рамках надежной среды передачи данных.

1. ОПИСАНИЕ И КЛАССИФИКАЦИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

Беспилотные летательные аппараты – это летательные аппараты, представляющие собой автономные роботизированные системы, задачей которых является выполнение полетов, потенциально опасных для человека, по заранее заданной программе с возможностью автоматической или ручной корректировки полетного задания, а также оперативное принятие решений в зависимости от меняющихся условий полета и окружающего пространства [1].

В наши дни классификация беспилотных летательных аппаратов (или дронов) в России ведется в соответствии с ГОСТ Р 59517–2021 «Беспилотные авиационные системы. Классификация и категоризация» [2]. Этот стандарт позволяет классифицировать БПЛА по четырем параметрам и разделяет их на три категории.

Однако в вопросах компоновки БПЛА производители не ограничены никакими стандартами, в результате чего отсутствуют требования к оснащению БПЛА со стороны авиационных регуляторов. Возможно это связано с тем, что БПЛА уже имеют множество конфигураций, аэродинамических схем и их компонентов, а также с тем, что внедрение жестких стандартов и классификаций может ограничить развитие технологий БПЛА.

Таким образом, на основании ГОСТ Р 59517–2021 БПЛА могут классифицироваться по следующим параметрам:

1) максимальная взлетная масса;
2) достигаемая в полете кинетическая энергия;
3) эксплуатационное назначение;
4) условия видимости.

Беспилотные летательные аппараты разделяются на три категории:

1) открытая категория (А);
2) специальная категория (В);
3) сертифицируемая категория (С).

Согласно российской классификации БПЛА можно систематизировать следующим образом [3]:

– микро- и мини-БПЛА ближнего радиуса действия (взлетная масса до 5 кг, дальность действия 25…40 км);

– легкие БПЛА малого радиуса действия (взлетная масса 5…50 кг, дальность действия 10…70 км);

– легкие БПЛА среднего радиуса действия (взлетная масса 50…100 кг, дальность действия 70…150 (250) км);

– средние БПЛА (взлетная масса 100…300 кг, дальность действия 150…1000 км);

– среднетяжелые БПЛА (взлетная масса 300…500 кг, дальность действия 70…300 км);

– тяжелые БПЛА среднего радиуса действия (взлетная масса более 500 кг, дальность действия 70…300 км);

– тяжелые БПЛА большой продолжительности полета (взлетная масса более 1500 кг, дальность действия около 1500 км);

– беспилотные боевые самолеты (взлетная масса более 500 кг, дальность действия около 1500 км).

Международной ассоциацией по беспилотным системам AUVSI (Association for Unmanned Vehicle Systems International) была предложена универсальная классификация БПЛА, которая объединяет в себе вышеназванные критерии [4].

Универсальная классификация БПЛА по летным параметрам

Universal classification of UAVs by flight parameters

Универсальная классификация БПЛА по летным параметрам						
Группа	Категория		Взлетная масса, кг	Дальность полета, км	Высота полета, м	Продолжительность полета, ч
	рус.	англ.				
Малые БПЛА	НаноБПЛА	Nano	< 0,025	< 1	100	1
	МикроБПЛА	Micro	<5	< 10	250	1
	Мини-БПЛА	Mini	5…150	< 10	150… 300	<2
Тактические	Легкие БПЛА для контроля переднего края обороны	Close Range (CR)	25…150	10…30	3000	2…4
	Легкие БПЛА с малой дальностью полета	Short Range (SR)	50…250	30…70	3000	3…6

Продолжение таблицы

Continuation of the Table

Универсальная классификация БПЛА по летным параметрам						
Группа	Категория		Взлетная масса, кг	Дальность полета, км	Высота полета, м	Продолжительность полета, ч
	рус.	англ.				
	Средние БПЛА	Medium Range (MR)	150…500	70…200	5000	6…10
	Средние БПЛА с большой продолжительностью полета	Medium Range Endurance (MRE)	500…1500	>500	8000	10…18
	Маловысотные БПЛА для проникновения в глубину обороны	Low Altitude Deep Penetration (LADP)	250…2500	>250	50… 9000	0,5…1
	Маловысотные БПЛА с большой продолжительностью полета	Low Altitude Long Endurance (LALE)	15…25	>500	3000	>24
	Средневысотные БПЛА с большой продолжительностью полета	Medium Altitude Long Endurance (MALE)	1000…1500	>500	5000… 8000	24…48
Стратегические	Высотные БПЛА с большой продолжительностью полета	High Altitude Long Endurance (HALE)	2500…5000	>2000	20 000	24…48
Спецназначение	БПЛА, оснащенные ударной частью	Lethal (LET) (Offensive)	–	300	4000	3…4
	БПЛА – ложные цели	Decoys (DEC)	150…500	0…500	50…5000	<4

Окончание таблицы

End of the Table

Универсальная классификация БПЛА по летным параметрам						
Группа	Категория		Взлетная масса, кг	Дальность полета, км	Высота полета, м	Продолжительность полета, ч
	рус.	англ.				
	Стратосферные БПЛА	Stratospheric (STRA)	>2500	>2000	>20 000	>48
	Экзостратосферные БПЛА	Exo-stratospheric (EXO)	–	–	>30 500	–

Российская классификация отличается от предложенной UVS International по ряду параметров: некоторые классы зарубежной классификации отсутствуют в РФ, легкие БПЛА в России имеют значительно большую дальность и т. д.

2. БАЗОВЫЕ СИСТЕМЫ БПЛА

Технологии развития БПЛА находятся на стыке прикладных наук и высокотехнологичных отраслей. Создание аэродинамических схем летательных аппаратов, разработка специальных материалов для изготовления конструкций фюзеляжа, изготовление нанотехнологичных процессоров, сенсоров – все эти направления должны слаженно работать, чтобы создаваемые БПЛА могли отвечать всем предъявляемым к ним требованиям. В настоящее время уровни технологического и промышленного развития, материаловедения, а также развития цифровых технологий позволили создавать автономные высокоточные БПЛА с высокими летно-техническими и массогабаритными характеристиками. При изготовлении БПЛА используются композитные материалы, которые позволяют значительно повысить маневренность и прочность, уменьшить вес производимого БПЛА, способствуют поглощению вибраций и уменьшению производимого уровня шума при полете. Характеристики материалов, из которых создаются БПЛА, при различных параметрах окружающей среды позволяют БПЛА совершать полеты на больших высотах и с высоким уровнем перегрузки практически в любую погоду. Основным и самым высокотехнологичным элементом системы БПЛА является электронная система управления.

Электронная система управления любого БПЛА состоит из вычислительной мощности и сенсоров, включающих в себя следующее:

1) процессор с модулями оперативной и энергонезависимой памяти, необходимые для функционирования систем БПЛА;

2) модуль определения положения в пространстве, состоящий из многоосевых MEMS-сенсоров, таких как магнетометры, акселерометры, гироскопы и т. д.;

3) модуль аналоговых или цифровых барометрических датчиков для определения высоты и воздушной скорости;

4) модуль управления двигателями и энергоснабжением;

5) модуль управления сервоприводами для управления полетом и режимами двигателей;

6) модуль приема спутниковой навигации GPS для точного геопозиционирования;

7) модуль радиосвязи для ручного управления и передачи данных телеметрии.

Систему управления БПЛА также могут дополнять другие системы, например:

– радиолокационные системы;
– лидары;
– ультразвуковые датчики расстояний;
– системы стабилизации фото- и видеооборудования.

Устройство стандартного БПЛА вертолетного типа (квадрокоптер) и его основные узлы представлены на рис. 1.

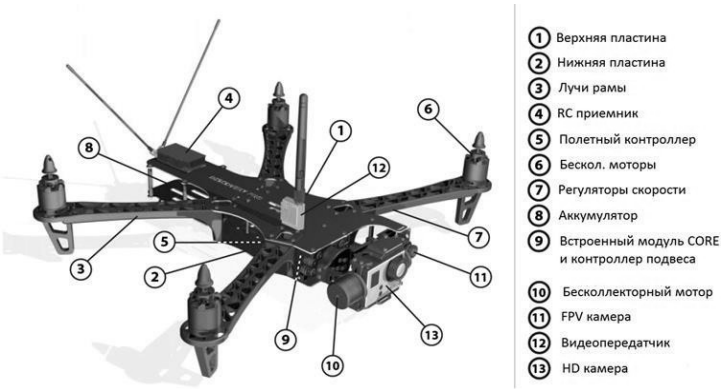

Рис. 1. Устройство стандартного БПЛА

Fig. 1. The structure of a standard UAV

3. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ КФС

Концептуальная модель киберфизической системы [6] включает в себя пять уровней:

1) физический;
2) сетевой;
3) хранилище данных;
4) обработка и аналитика;
5) уровень приложений.

На рис. 2 представлена концептуальная модель киберфизической системы.

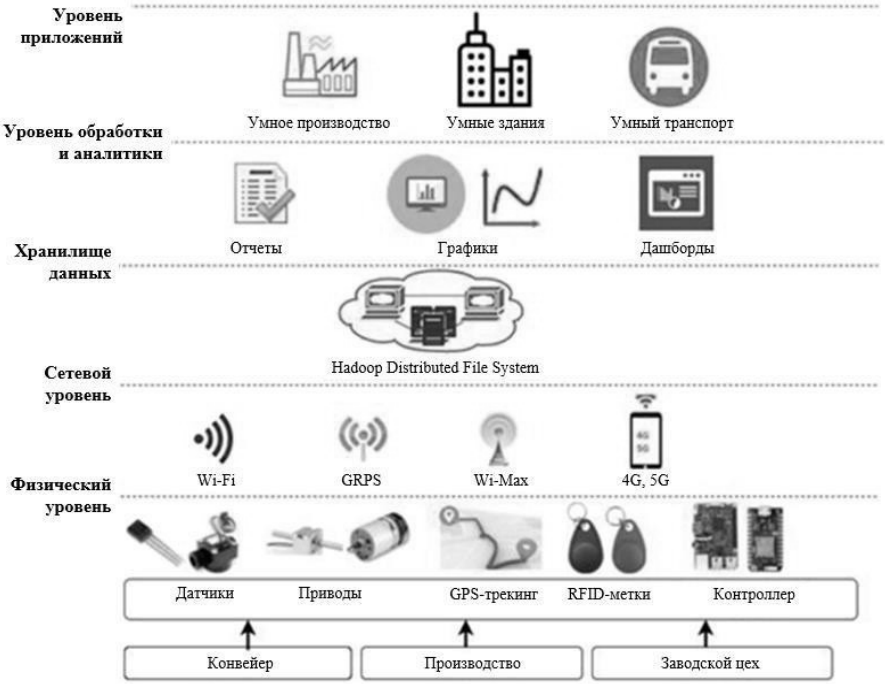

Рис. 2. Концептуальная модель киберфизической системы

Fig. 2. Conceptual model of a cyberphysical system

Физический уровень

Этот уровень состоит из приводов, датчиков, вычислительных элементов и отслеживающих устройств. В режиме реального времени контроллер спосо-

бен собирать с датчиков данные и обрабатывать их локально и/или передавать в облачное хранилище для обработки.

Сетевой уровень

Киберфизические системы могут получать доступ к киберпространству по различным сетевым протоколам, например:

– Wi-Fi,
– WiMAX,
– GPRS,
– технологии 3G/4G/LTE.

Другие облегченные протоколы, такие как MQTT, CoAP, AMQP, Websocket и Node, используются для передачи данных с периферийных устройств в облако для их хранения и обработки [6]. Каждый протокол имеет свои плюсы перед другими в зависимости от пропускной способности, скорости, надежности, задержки, масштабируемости и безопасности.

Хранилище данных

Киберфизические системы обрабатывают большой объем данных, собранных с объектов, расположенных на физическом уровне. Такие данные хранятся на локальном сервере или в облаке. Данные могут храниться в кластере из трех различных подчиненных узлов с использованием распределенной системы хранения для резервирования

Уровень обработки и аналитики

Уровень обработки и аналитики применяется для обработки данных. С использованием SQL-запросов можно генерировать отчеты, графики, осуществлять мониторинг в режиме реального времени. Методы интеллектуального анализа данных, такие как кластеризация данных, классификация и регрессия, могут быть использованы для планирования и построения прогнозов. Также на этом уровне процессы мониторинга и управления могут быть направлены обратно на физический уровень для приведения в действие некоторых устройств.

Уровень приложений

Этот уровень является пользовательским интерфейсом для операторов, производителей, сторонних поставщиков. Также приложения имеют удобный интерфейс, позволяющий взаимодействовать с уровнями КФС на основе привилегированного доступа и приоритета.

4. ПРИМЕНЕНИЕ КИБЕРФИЗИЧЕСКИХ СИСТЕМ

В настоящее время применение КФС охватывает большинство производственных сфер. Но наиболее востребованными они являются в области строительства, транспорта, производства и энергосбережения. Например, умная

электрическая сеть (Smart Grid) способна объединять несколько производственных электростанций с множеством нагрузок и в режиме реального времени производить динамическую балансировку нагрузки и ценообразование. Киберфизические системы в сфере производства представлены умными станками и машинами, технологиями 3D-печати и аддитивного производства в целом. Если роботы до последнего времени были оправданы на производстве только для стандартизированных повторяющихся операций, то гибридные системы, способные адаптироваться к изменениям, могут значительно расширить область действий автоматизированных систем.

На рис. 3 представлены сферы, в которых уже успешно применяются киберфизические системы.

Рис. 3. Сферы применения киберфизических систем

Fig. 3. Application areas of cyber-physical systems

5. КЛАССИФИКАЦИЯ КФС

В настоящее время сложно обозначить единую классификацию киберфизических систем. В обобщенном виде основные атрибуты (признаки) классификации КФС можно представить следующим образом.

1. Сложность в соответствии с функциональными возможностями и используемыми компонентами.

2. Связность в соответствии с используемыми интерфейсами и протоколами передачи данных.

3. Критичность в соответствии с зависящими от системы бизнес-процессами.

4. Социальный аспект в соответствии с характером взаимодействия системы с пользователями и операторами.

Эти атрибуты в большей степени применимы к концепции рассмотрения БПЛА как киберфизической системы. Именно по этой причине в дальнейшем будем использовать вышеуказанную классификацию.

Рассмотрение БПЛА как КФС позволяет определять их как совокупность датчиков, систем, вычислительных мощностей, а не как отдельный, неделимый объект. Подобная концепция в дальнейшем даст возможность определять элементы подобной КФС как уязвимости (точки входа) и при рассмотрении БПЛА использовать парадигмы, применимые к киберфизическим системам.

6. БПЛА КАК КИБЕРФИЗИЧЕСКАЯ СИСТЕМА

БПЛА является многосоставной системой и может включать в себя тысячи различных датчиков и сенсоров, которые обмениваются между собой информацией, аккумулируют ее и передают на управляющее устройство. Одиночные БПЛА или группу БПЛА можно рассматривать как информационную систему со своими структурно-функциональными характеристиками.

В БПЛА встроена киберфизическая система, состоящая из датчиков и/или исполнительных механизмов. Датчики предоставляют данные (или информацию) исполнительному механизму, который может управлять БПЛА. Собранные данные используются для анализа и принятия важных решений, связанных с полетной миссией.

Исходя из вышеописанного беспилотные летательные аппараты можно характеризовать как сложную систему, выполняющую сбор, хранение, обработку данных, поступающих от физических устройств (сенсоров, датчиков и т. д.), а также их тесную интеграцию с информационными технологиями в рамках надежной среды передачи данных (каналы управления и навигации).

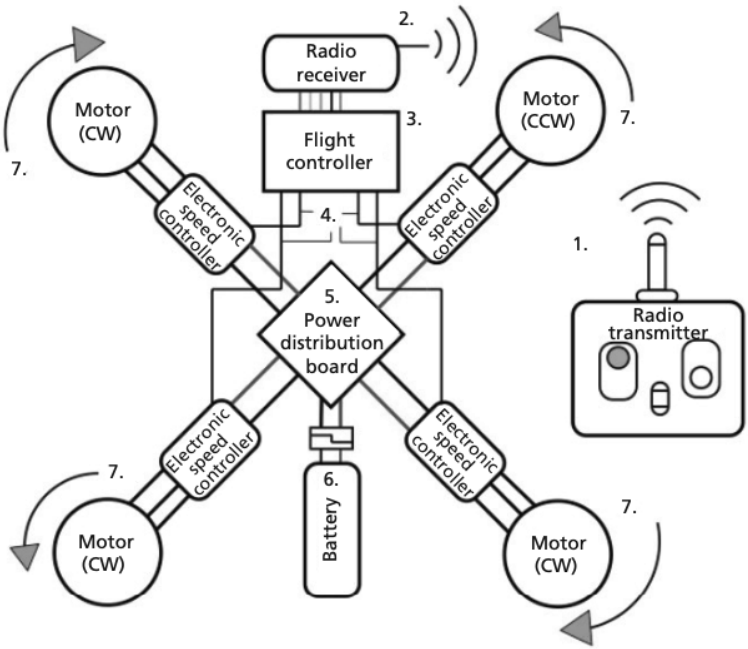

Рис. 4. Используемая в исследованиях структурная схема БПЛА [8]

Fig. 4. Structural diagram of the UAV used in the research [8]

Таким образом, в ряде случаев БПЛА можно рассматривать как киберфизическую систему, что позволит в дальнейшем применять к БПЛА ряд определений, свойственных КФС, и методики исследования информационной безопасности, в том числе модели угроз, модели атак, разрабатываемые для КФС.

7. СРАВНЕНИЕ КИБЕРНЕТИЧЕСКОЙ И КИБЕРФИЗИЧЕСКОЙ МОДЕЛИ KILL-CHAIN

После того как мы представили БПЛА в виде киберфизической системы, мы можем рассматривать и применять модели кибератак для киберфизических систем непосредственно к БПЛА.

Как один из примеров, рассмотрим сравнение кибернетической и киберфизической модели kill-chain, которая представлена на рис. 5 [7].

Рис. 5. Сравнение кибернетической и киберфизической модели kill-chain

Fig. 5. Comparison of cybernetical and cyberphysical kill-chain models

В соответствии с типовой моделью kill-chain для киберсистем все этапы атаки проводятся на одном уровне – физическом. Модель kill-chain для киберфизической системы организована на трех различных уровнях – физическом, уровне управления и киберуровне.

После определения БПЛА как киберфизической системы мы можем рассматривать различные модели кибератак (и модели защиты от них), соответствующие киберфизическим системам.

8. КИБЕРУГРОЗЫ БЕЗОПАСНОСТИ БПЛА. ИСТОЧНИКИ И РЕАЛИЗАЦИЯ

Чаще всего взлом производится с целью перехвата или подмены информации, передаваемой с помощью БПЛА. Современное ПО позволяет существенно снизить риск ошибок, а также повысить уровень автономности. Однако БПЛА имеет уязвимые места за счет факторов, оказывающих пагубное влияние на всю систему в целом.

Рассмотрим некоторые из них:

1) разрушающее радиоэлектронное воздействие на информационно-управляющую систему;

2) несанкционированный доступ к основным узлам на программном уровне и, как следствие, нарушение технологических циклов работы;

3) нарушение (срыв) управления из-за деструктивного воздействия вредоносного программного обеспечения (ПО);

4) человеческий фактор (свободный доступ к элементам БРЭО, ошибки программистов);

5) использование штатных операционных систем и аппаратных средств с имеющимися недекларированными возможностями.

Но самым слабым местом любого БПЛА являются беспроводные каналы связи. Любые сигналы и команды, посылаемые GPS-навигатором, можно перехватывать, заглушить или подменить.

Атаки на БПЛА через уязвимость в канале связи можно разделить на три категории:

1) усиление или создание радиопомех, запуск в систему стороннего вредоносного ПО;

2) перехват трафика;

3) имитация и подмена сигналов GPS.

Исходя из вышеизложенного можно сделать вывод, что беспилотные летательные аппараты в дальнейшем можно позиционировать как киберфизические системы, а также применять к БПЛА угрозы безопасности, относящиеся к КФС.

СПИСОК ЛИТЕРАТУРЫ

1. Беспилотные летательные аппараты // Российские беспилотники. – URL: https://russiandrone.ru/publications/bespilotnye-letatelnye-apparaty/ (дата обращения: 30.08.2024).

2. ГОСТ Р 59517–2021. Беспилотные авиационные системы. Классификация и категоризация. – М.: Стандартинформ, 2021.

3. Российская универсальная классификация // Арсенал-Инфо. – URL: https://arsenal-info.ru/b/book/3398882726/15 (дата обращения: 30.08.2024).

4. Классификация UVS International // Арсенал-Инфо. – URL: https://arsenal-info.ru/b/book/3398882726/14 (дата обращения: 30.08.2024).

5. NSF Workshop on Cyber-Physical Systems, October 16–17, 2006. – Austin, Texas, 2006.

6. *Смышляева А.А., Резникова К.М., Савченко Д.В.* Современные технологии в Индустрии 4.0 – киберфизические системы // Отходы и ресурсы. – 2020. – Т. 7, № 3. – Ст. 2. – DOI: 10.15862/02INOR320.

7. A multi-layered and kill-chain based security analysis framework for cyber-physical systems / A. Hahn, R.K. Thomas, I. Lozano, A. Cardenas // International Journal of Critical Infrastructure Protection. – 2015. – Vol. 11. – P. 39–50. – DOI: 10.1016/j.ijcip.2015.08.003.

8. How to analyze the cyber threat from drones: background, analysis frameworks, and analysis tools / K. Best, J. Schmid, S. Tierney, J. Awan, N. Beyene, M. Holliday, R. Khan, K. Lee. – RAND Corporation, 2020. – DOI: 10.7249/RR2972.

Васильев Евгений Алексеевич, аспирант кафедры безопасности информационных технологий. Основное направление научных исследований – безопасность БПЛА, методы радиоэлектронной борьбы. E-mail: evva@sfedu.ru

Абрамов Евгений Сергеевич, заведующий кафедрой безопасности информационных технологий. Основное направление научных исследований – обнаружение атак, методы компьютерной криминалистики. E-mail: abramoves@sfedu.ru

DOI: 10.17212/2782-2230-2024-3-63-77

UAV as a cyberphysical system *

E.A. Vasiliev[1], E.S. Abramov[2]

[1] *Institute of Computer Technologies and Information Security, ITA SFedU, 2 Chekhova Street, Taganrog, 347922, Russian Federation, postgraduate student of the Department of Information Technology Security. E-mail: evva@sfedu.ru*

[2] *Institute of Computer Technologies and Information Security, ITA SFedU, 2 Chekhova Street, Taganrog, 347922, Russian Federation, Head of the department of the Department of Information Technology Security. E-mail: abramoves@sfedu.ru*

Currently, the scientific community gives us separate definitions of UAVs and cyberphysical systems. These definitions are completely self-sufficient and do not intersect with each other. However, technical progress does not stand still and the narrowness of these definitions does not allow researchers to fully apply some definitions due to problems in terminology. This article will help overcome the existing differences in terminology and transfer the properties of cyber-physical systems to unmanned aerial vehicles.

Keywords: UAV, cyberphysical system, conceptual model, threat models, attack models, kill-chain model for CFS, UAV vulnerabilities, negative consequences

REFERENCES

1. Bespilotnye letatel'nye apparaty [Unmanned aerial vehicles]. *Rossiiskie bespilotniki* [Russian drones]. Available at: https://russiandrone.ru/publications/bespilotnye-letatelnye-apparaty/ (accessed 30.08.2024).

2. GOST R 59517–2021. *Bespilotnye aviatsionnye sistemy. Klassifikatsiya i kategorizatsiya* [State standard R 59517–2021. Unmanned aircraft systems. Classification and categorization]. Moscow, Standartinform Publ., 2021.

* *Received 18 August 2024.*

3. Rossiiskaya universal'naya klassifikatsiya [Russian universal classification]. *Arsenal-Info*. (In Russian). Available at: https://arsenal-info.ru/b/book/3398882726/15 (accessed 30.08.2024).

4. Klassifikatsiya UVS International [Classification UVS International]. *Arsenal-Info*. (In Russian). Available at: https://arsenal-info.ru/b/book/3398882726/14 (accessed 30.08.2024).

5. *NSF Workshop on Cyber-Physical Systems*, October 16–17, 2006, Austin, Texas.

6. Smyshlyaeva A.A., Reznikova K.M., Savchenko D.V. Sovremennye tekhnologii v Industrii 4.0 – kiberfizicheskie sistemy [Modern technologies in Industry 4.0 – cyber-physical systems]. *Otkhody i resursy = Russian Journal of Resources, Conservation and Recycling*, 2020, vol. 7, no. 3, art. 2. DOI: 10.15862/02INOR320.

7. Hahn A., Thomas R.K., Lozano I., Cardenas A. A multi-layered and kill-chain based security analysis framework for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 2015, vol. 11, pp. 39–50. DOI: 10.1016/j.ijcip.2015.08.003.

8. Best K., Schmid J., Tierney S., Awan J., Beyene N., Holliday M., Khan R., Lee K. *How to analyze the cyber threat from drones: background, analysis frameworks, and analysis tools*. RAND Corporation, 2020. DOI: 10.7249/RR2972.

Для цитирования:

Васильев Е.А., Абрамов Е.С. БПЛА как киберфизическая система // Безопасность цифровых технологий. – 2024. – № 3 (114). – С. 63–77. – DOI: 10.17212/2782-2230-2024-3-63-77.

For citation:

Vasiliev E.A., Abramov E.S. BPLA kak kiberphizicheskaya sistema [UAV as a cyber-physical system]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 3 (114), pp. 63–77. DOI: 10.17212/2782-2230-2024-3-63-77.

ПРАВИЛА ДЛЯ АВТОРОВ

УСЛОВИЯ ПРИЕМА СТАТЕЙ

Все статьи и сопровождающие их материалы в журнал подаются через сайт журнала в электронном виде после регистрации всех авторов статьи. Регистрация обязывает каждого автора иметь международный идентификационный номер ORCID. Иные варианты подачи материалов не рассматриваются.

Автор (один из соавторов) в своем личном кабинете выбирает в меню пункт «Подать статью» и вводит все необходимые данные. Своих соавторов при этом он выбирает из списка зарегистрированных пользователей.

Рукопись статьи готовится в соответствии с правилами оформления в редакторе MS Word и прикрепляется в формате *.doc, *.docx.

Сканированные лицензионный договор с подписями авторов и экспертное заключение (цветной режим сканирования, разрешение не менее 600 dpi) необходимо также разместить на сайте журнала в разделе «Подать статью» в формате *.pdf, *.jpg, *.jpeg.

По окончании всех работ обязательно нажать кнопку «Отправить в редакцию».

В редакцию журнала представляются следующие материалы.

1. **Статья**, подготовленная в соответствии с правилами оформления, – печатная версия, 2 экземпляра, подписанных авторами.

2. **Контактная информация** (телефоны рабочий и сотовый, адреса электронной почты, место работы, адрес места работы, должность, ученая степень, ученое звание автора) – печатная версия, 2 экземпляра.

3. **Описание статьи для базы данных «Российский индекс научного цитирования (РИНЦ)»**, подготовленное в соответствии с правилами оформления, – печатная версия, один экземпляр.

4. **Лицензионный договор**, заполненный и подписанный.

5. **Электронная версия статьи, контактной информации, описания статьи для базы данных РИНЦ, сканированный лицензионный договор и экспертное заключение о возможности опубликования отправляются отдельными файлами на адрес редакции**.

6. **Согласие на публикацию, обработку и распространение персональных данных авторов статей**.

7. **Экспертное заключение о возможности опубликования**.

Редакцией рассматриваются только те материалы авторов, которые полностью соответствуют вышеобозначенным требованиям. Неполный пакет материалов редакцией не рассматривается.

Подготовленные материалы направляются на почтовый адрес редакции: 630073, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет (НГТУ), корп. 7, ком. 606, в редакцию журнала «Безопасность цифровых технологий».

Все рукописи рецензируются, по результатам рецензирования редколлегия принимает решение о целесообразности опубликования материалов.

ВНИМАНИЕ!

Авторы несут ответственность за оформление, содержание и сам факт публикации статьи. Редакция журнала не несет ответственности за возможный ущерб, вызванный публикацией статьи. При наличии существенных недостатков в оформлении и содержании статьи редакция принимает решение об отклонении статьи без приведения полного перечня ошибок автора.

Ранее опубликованные материалы, а также материалы, представленные для публикации в других журналах, к рассмотрению не принимаются.

ПРАВИЛА ОФОРМЛЕНИЯ

При подготовке документов для отправки в редакцию журнала авторам рекомендуется внимательно прочитать правила и посмотреть примеры оформления статей и всех необходимых сопутствующих документов. Редакция рассматривает статьи, подготовленные как на русском, так и на английском языке. Для опубликования статьи на английском языке необходимо дополнительно предоставить ее русскоязычный вариант, оформленный по правилам журнала (кроме зарубежных авторов).

Перед отправкой рукописи в редакцию авторам необходимо проверить свою статью с помощью системы «Антиплагиат». Принятый редакционной коллегией уровень оригинальности статей должен составлять не менее 85 %.

Чтобы статья была направлена на рецензирование, необходимо подготовить следующее:

1) **статью** в соответствии с правилами оформления;

2) **контактную информацию** в одном файле предоставить по каждому автору: ФИО полностью, ученая степень, ученое звание, должность, место работы, адрес места работы, телефон рабочий и мобильный, адрес электронной почты;

3) **описание статьи для базы данных «Российский индекс научного цитирования (РИНЦ)»**;

4) **лицензионный договор** заполнить, бланк лицензионного договора должен быть подписан только авторами (он доступен авторам также в личном кабинете). Если авторов несколько, то необходимо добавить поля на всех авторов и подписать каждому из них;

5) **экспертное заключение** о возможности опубликования, принятое в вашей организации;

6) согласие на публикацию, обработку и распространение персональных данных авторов статей;

7) авторы, не являющиеся сотрудниками НГТУ, предоставляют **сопроводительное письмо** на имя проректора по научной работе НГТУ (ссылка на страницу сайта НГТУ). Письмо нужно подготовить на бланке организации с подписью и печатью руководителя.

ОСНОВНЫЕ РАЗДЕЛЫ ЖУРНАЛА

Автоматизация и управление технологическими процессами и производствами.

Управление в социальных и экономических системах.

Методы и системы защиты информации, информационная безопасность.

RULES FOR AUTHORS

CONDITIONS FOR ACCEPTANCE OF ARTICLES

All articles and their accompanying materials are submitted to the magazine through the magazine's website in electronic form after registration of all the authors of the article. Registration obliges each author to have an international ORCID. No other material supply options are considered.

The author (one of the co-authors) in his personal account selects the item "Submit article" in the menu and enters all the necessary data. At the same time, he selects his co-authors from the list of registered users.

The manuscript of the article is prepared in accordance with the design rules in the MS Word editor and attached in the format * .doc, * .docx.

Scanned license agreement with signatures of authors and expert opinion (color mode scanning, resolution not less than 600 dpi) can also be attached on the website of the magazine in the section "Submit article" in the format * .pdf, * .jpg, * .jpeg.

At the end of all works, be sure to click the "Send to Design" button.

The following materials are provided to the journal editor:

1. **The article**, prepared in accordance with the rules of design, is a private version, 2 copies signed by the authors.
2. **Contact information** (working and cellular phones, e-mail addresses, place of work, address of the place of work, position, scientific degree, academic title of the author) – printed version, 2 copies.
3. **The description of the article** for the database "**Russian Scientific Citation Index (RSCI)**", prepared in accordance with the rules of form-making, is a printed version, one copy.
4. **License agreement** completed and signed.
5. **Electronic version of the article**, contact information, description of the article for the RSCI database, scanned license agreement and expert opinion on the possibility of publication (in separate files to the editorial address).
6. **Consent to the publication, processing and dissemination of the personal data** of the authors of the articles.
7. **Expert opinion** on the possibility of publication.

The editors consider only those materials of the authors that fully meet the above requirements. Incomplete package of materials is not considered by the revision.

The prepared materials are sent to the postal address of the editorial office: 630073, Novosibirsk, Karl Marx Prospekt, 20, Novosibirsk State Technical University (NSTU), building 7, office 606, to the editors of the journal "Digital Technology Security".

All manuscripts were reviewed, and according to the results of the review, the editorial board decided on the appropriateness of publishing the materials.

ATTENTION!

The authors are responsible for the design, content and the fact of publication of the article. The editorial board of the journal is not responsible for possible damage caused by the publication of the article. If there are significant shortcomings in the design and content of the article, the editorial board decides to reject the article without giving a full list of the author's mistakes.

Previously published materials, as well as materials submitted for publication in other journals, are not accepted for consideration.

FORMATTING RULES

When preparing documents for submission to the journal editor, authors are advised to carefully read the rules and see examples of the design of articles and all necessary related documents. The Drafting Committee considered articles prepared in both Russian and English. To publish the article in English, it is necessary to additionally provide its Russian-language version, drawn up according to the rules of the magazine (except for foreign authors).

Before sending the manuscript to the editorial office, authors must check their article using the Antiplagiarism system. The level of originality of articles adopted by the Editorial Board should be at least 85 %.

For the article to be aimed at peer review, you need to prepare the following:

1) **the article** in accordance with the rules of design (volume from 7 to 30 pages);

2) **provide contact information** in one file for each author: full name, degree, academic title of the author, position, place of work, address of the place of work, telephone number of the worker and mobile, e-mail address;

3) **description of the article** for the database "Russian Scientific Citation Index (RSCI)";

4) fill out the **license agreement**, the form of the license agreement must be signed only by the authors (it is also available to the authors in the personal office), if there are several authors, then it is necessary to add fields on all authors and sign each of them;

5) **expert opinion** on the possibility of publication, adopted in your organization;

6) consent to the publication, processing and dissemination of the personal data of the authors of the articles;

7) authors who are not employees of the NSTU provide a **companion letter** addressed to the vice-rector for scientific work of the NSTU (link to the page of the NSTU website). The letter should be prepared on the form of the organization with the signature and seal of the manager.

JOURNAL SECTION

Automation and control of technological processes and productions.
Governance in social and economic systems.
Methods and systems of information protection, information security.