Научная статья
УДК 004.056
DOI:10.31854/1813-324X-2023-9-1-105-115

Иерархическая модель для проектирования систем на основе микроконтроллеров защищенными от киберфизических атак

Дмитрий Сергеевич Левшун, levshun@comsec.spb.ru

[1]Санкт-Петербургский Федеральный исследовательский центр Российской академии наук,
Санкт-Петербург, 199178, Российская Федерация

Аннотация: *В статье предложена иерархическая модель для проектирования систем на основе микроконтроллеров защищенными от киберфизических атак. В рамках данной модели система на основе микроконтроллеров представляется в виде иерархического реляционного множества взаимодействующих блоков с различными свойствами и связями между ними. Предложенная модель включает в себя модели аппаратных, программных и программно-аппаратных элементов, интерфейсов, протоколов и связей между элементами системы на различных ее уровнях, модели атакующего и атакующих действий. Ключевое отличие разработанной модели заключается в возможности представления систем на основе микроконтроллеров в целом, что позволяет нивелировать основной недостаток аналогов – проектирование систем на основе микроконтроллеров без учета взаимодействия их устройств друг с другом. Кроме того, разработанная модель является модульной и расширяемой, направлена на обеспечение защищенности проектируемого решения от киберфизических атак, а также рассматривает элементы защиты как неотъемлемую часть спроектированной системы.*

Ключевые слова: *информационная безопасность, проектирование защищенных систем, система на основе микроконтроллеров, теоретико-множественная модель, киберфизическая атака, иерархическая модель*

Источник финансирования: Исследование выполнено за счет гранта Российского Научного Фонда № 22-71-00107, https://rscf.ru/project/22-71-00107.

Ссылка для цитирования: Левшун Д.С. Иерархическая модель для проектирования систем на основе микроконтроллеров защищенными от киберфизических атак // Труды учебных заведений связи. 2023. Т. 9. № 1. С. 105–115. DOI:10.31854/1813-324X-2023-9-1-105-115

Hierarchical Model for the Design of Microcontroller-Based Systems Protected from Cyber-Physical Attacks

Dmitry Levshun, levshun@comsec.spb.ru

[1]Saint-Petersburg Federal Research Center of the Russian Academy of Sciences,
St. Petersburg, 199178, Russian Federation

Abstract: *The article proposes a hierarchical model for the design of microcontroller-based systems protected from cyber-physical attacks. Within the framework of this model, a microcontroller-based system is represented as a hierarchical relational set of interacting building blocks with different properties and links between them. The proposed model includes models of hardware, software and hardware-software elements, interfaces, protocols and*

links between system elements, models of attacker and attack actions. The key difference of the developed model lies in the possibility of full representation of microcontroller-based systems, while other solutions have a drawback of designing such systems without taking into account the interactions of their devices with each other and other systems. In addition, the developed model is modular and extensible, aims to ensure the security of the designed solution from cyber-physical attacks, and considers security elements as an integral part of the final solution.

Keywords: *information security, secure by design, microcontroller-based system, set-theoretic approach, cyber-physical attack, hierarchical model*

Funding: The study was supported by the grant of the Russian Science Foundation No. 22-71-00107, https://rscf.ru/en/project/22-71-00107.

For citation: Levshun D. Hierarchical Model for the Design of Microcontroller-Based Systems Protected from Cyber-Physical Attacks. *Proc. of Telecom. Universities.* 2023;9(1):105–115. (in Russ.) DOI:10.31854/1813-324X-2023-9-1-105-115

1. Введение

В настоящее время системы на основе микроконтроллеров – неотъемлемая часть любой сферы жизнедеятельности человека, что обусловливает критическую важность обеспечения их защищенности [1]. Последствия отказа подобных систем, в том числе связанные с деятельностью злоумышленников, включают в себя как финансовый и репутационный ущерб, так и угрозу жизни и здоровью человека. Одним из возможных направлений атаки является использование уязвимостей, наличие которых в подобных системах обусловлено различными факторами.

Уязвимости, возникающие из-за ошибок на этапе проектирования, являются наиболее опасными, т. к. их устранение, как правило, представляет собой трудно решаемую задачу [2]. Особенно когда устранение ошибки подразумевает изменения в аппаратной или программной составляющих отдельных устройств, в то время как их фирм-производителей уже не существует. Распространенность таких уязвимостей связана с тем, что системы на основе микроконтроллеров зачастую проектируются без участия специалистов в области информационной безопасности с применением слабозащищенных протоколов передачи данных, выходом в сеть Интернет и использованием непроверенного на наличие ошибок кода [3].

Например, согласно отчету компании SonicWall, количество атак с помощью устройств на основе микроконтроллеров подскочило на 215,7 %: до 32,7 миллионов в 2018 г. (по сравнению с 10,3 миллионами в 2017 г.). В 2019 г. атаки продолжились, но, согласно отчету данной компании, их количество увеличились только на 5 %. Также, согласно отчету «Unit 42 Threat Report» компании Palo Alto Networks за 2020 г., «98 % трафика устройств не зашифровано, что открывает нарушителям доступ к личным и конфиденциальным данным их пользователей».

Решение данной проблемы является важной задачей, именно поэтому были разработаны и применяются на практике различные методики проектирования [4]. При этом одним из ключевых элементов любого подхода к проектированию систем на основе микроконтроллеров является используемая в них модель системы.

В отличие от существующих решений, предлагаемая в данной работе расширяемая иерархическая, основанная на множествах, реляционная модель позволяет описывать системы на основе микроконтроллеров, а не отдельные устройства данных систем. Данная особенность позволяет нивелировать один из недостатков аналогов – проектирование устройств на основе микроконтроллеров без учета их взаимодействия друг с другом.

Кроме того, разработанная модель, будучи модульной, расширяемой и иерархической, направлена на обеспечение защищенности проектируемого решения, а также рассматривает элементы защиты как неотъемлемую часть проектируемой системы. Расширение данной модели возможно за счет введения новых уровней абстракции. Модульность решения позволяет изменять отдельные части модели без необходимости полной ее переработки. Например, могут быть изменены параметры модели злоумышленника или анализируемые классы атакующих действий. Иерархический характер решения позволяет осуществлять прямые (от всей системы к отдельным элементам) и обратные (от отдельного элемента к системе в целом) переходы в рамках модели.

Статья организована следующим образом. В разделе 2 проанализированы существующие решения в области моделирования систем на основе микроконтроллеров. В разделе 3 предложена новая иерархическая модель для проектирования систем на основе микроконтроллеров защищенными от киберфизических атак. В разделе 4 показаны взаимосвязи между элементами представленной модели. В разделе 5 показаны достоинства, недостатки и область применения предложенной модели. В разделе 6 содержатся основные выводы и направления дальнейших исследований.

2. Анализ существующих решений

В современных исследованиях, для отображения различных аспектов сложных систем и анализа возможности реализации различных атакующих действий используется компонентное [5–7], полунатурное [8–10], имитационное [11–13] и аналитическое [14–16] моделирование. При этом каждый подход к моделированию имеет свой уровень полноты и детализации, а также назначение при представлении системы (таблица 1).

ТАБЛИЦА 1. Сравнение подходов к моделированию

TABLE 1. Comparison of Modeling Approaches

Полнота	Уровень детализации	Подход к моделированию	Атаки
↑	↓	Аналитическая	Многошаговые на систему в целом
		Имитационная	Затрагивающие ряд устройств системы
		Полунатурная	Затрагивающие отдельные устройства системы
		Компонентная	Затрагивающие отдельные элементы системы

Компонентная модель является наиболее подробным способом представления систем на основе микроконтроллеров, однако требует много времени и усилий для описания больших систем. Более того, с помощью данной модели невозможно представить различные динамические процессы. С другой стороны, с помощью аналитического моделирования можно достаточно быстро представить систему целиком, однако уровень детализации будет совершенно иным. Рассмотрим несколько примеров работ.

Модель для проектирования отказоустойчивых систем на основе микроконтроллеров представлена в [17]. По мнению авторов, устойчивая система обладает тремя основными характеристиками: стабильность, защищенность и системность. Стабильность означает, что система на основе микроконтроллеров со временем всегда достигает предсказуемого состояния. Защищенность означает, что разработанная система способна обнаруживать киберфизические атаки и противодействовать им. Системность означает, что программные и аппаратные элементы системы интегрированы вместе. При этом модель, предложенная авторами, включает шесть уровней: 1) восприятие; 2) обработка; 3) анализ; 4) объединение решений; 5) оператор и его взаимодействие с системой. Для достижения системой состояния отказоустойчивости, необходимо обеспечить следующие характеристики: надежность, безотказность, согласованность, совместимость и интегрированность.

Возможность расширения мехатронных систем до киберфизических проанализированы в работе [18]. Авторы рассмотрели общие черты таких систем и подчеркнули возрастающую роль киберфизических систем в производственном секторе. Кроме того, авторы представили ключевые сложности в процессе проектирования таких систем, а именно необходимость комплексного и масштабируемого подхода. Данный подход необходим для предотвращения междисциплинарных конфликтов в процессе разработки киберфизических систем. При этом описание требования масштабируемости дано на основе внешних и внутренних взаимодействий, управления процессами, моделирования поведения, топологических отношений и функциональной совместимости. Предлагаемый авторами подход к проектированию включает семь основных этапов, а именно: определение границ системы, многоуровневое моделирование, моделирование взаимодействий, топологическое моделирование, семантическое взаимодействие, мультиагентное моделирование и моделирование взаимодействий. В качестве примера авторы использовали процесс изготовления планшетов.

В [19] авторы определили требования к защищенному, надежному и отказоустойчивому контроллеру SDN (*аббр. от англ.* Software-Defined Network [20]). Была проанализирована архитектура контроллеров этого типа, даны рекомендации по улучшению их защищенности. При этом атрибуты защищенности были разделены на три основные группы: архитектура, интерфейсы и сервисы. С точки зрения архитектуры, авторами были проверены следующие функциональные особенности: уровень изоляции процессов управления (приложения), способы разрешения конфликтов политик безопасности, использование защищенного хранилища данных, дублирование элементов контроллеров, надежность приложений.

Подход для проектирования систем шифрования и дешифрования видео в режиме реального времени предложен в [21]. Авторы также разработали и протестировали аппаратную реализацию системы. При этом параметры защищенности были протестированы на основе критериев Национального института стандартов и технологий (NIST, *аббр. от англ.* National Institute of Standards and Technology [22]). Основываясь на теоретическом анализе и экспериментальных результатах, авторы пришли к выводу о целесообразности применения разработанного подхода для проектирования и разработки новых защищенных систем видеосвязи.

Модель для проектирования защищенных и эффективных с точки зрения ресурсопотребления устройств на основе микроконтроллеров представлена в работе [23]. В ней авторы предложили подход к объединению отдельных алгоритмов и методик, реализующих разные функции средств

обеспечения информационной безопасности. При этом разработанные авторами модель и подход предлагается использовать как часть процесса проектирования систем на основе микроконтроллеров. В основе моделирования устройств лежит MARTE – UML-профиль для моделирования и анализа систем реального времени [24]. Данный профиль состоит из следующих частей:

– необработанные данные (спецификация устройства на основе микроконтроллеров, описание элементов безопасности);

– входные данные, извлеченные из необработанных данных (функциональные и нефункциональные свойства, модели злоумышленников и угроз, требования к защищенности, необходимые ресурсы и совместимость устройств);

– функции процесса конфигурации (фильтрация, проверка, анализ совместимости и компоновка на основе критериальной оптимизации).

Методика анализа защищенности сетевых информационных потоков в системах на основе микроконтроллеров представлена в [25]. По мнению авторов, основная проблема обеспечения защищенности систем такого типа заключается в их работе в потенциально враждебной среде, в то время как их вычислительные возможности сильно ограничены. При этом для анализа защищенности предлагается моделирование данных потоков на следующих основных уровнях: уровень электронных схем (аппаратные потоки); уровень прошивок, операционных систем и приложений (программные потоки); а также уровень взаимодействия устройств (сетевые потоки). При этом предлагаемая методика состоит из двух основных элементов: топологического анализа и анализа информационных потоков на основе политик безопасности. Топологический подход основан на выявлении всех элементов систем, лежащих между двумя узлами графа – источником и получателем данных. Подход на основе политик использует правила фильтрации трафика, конфигурацию сети и описание аномалий для обнаружения конфликтов в политиках безопасности. Данная методика была разработана в рамках проекта SecFutur [26].

Проведенный анализ позволяет отметить следующие недостатки существующих решений для моделирования и проектирования защищенных систем на основе микроконтроллеров:

1) учитываются только отдельные аспекты обеспечения защищенности систем на основе микроконтроллеров;

2) не рассматриваются сильные зависимости между программным и аппаратным обеспечением устройств на основе микроконтроллеров;

3) устройства на основе микроконтроллеров проектируются без учета параметров системы, элементов которой они являются;

4) защищенная связь между устройствами обеспечивается только внутри системы на основе микроконтроллеров, в то время как взаимодействие данных устройств с внешними системами не рассматривается.

Это означает, что единый подход к моделированию систем на основе микроконтроллеров для решения задачи их проектирования защищенными от киберфизических атак еще не сформирован и активно исследуется. В следующем разделе данной работы предложена оригинальная иерархическая модель, в рамках которой системы на основе микроконтроллеров представляются в виде иерархического реляционного множества взаимодействующих блоков с различными свойствами и связями между ними.

3. Системы на основе микроконтроллеров

В данной работе для представления систем на основе микроконтроллеров используется разделение их элементов на *компоненты*, *контроллеры* и *устройства*, взаимодействующие друг с другом (рисунок 1).

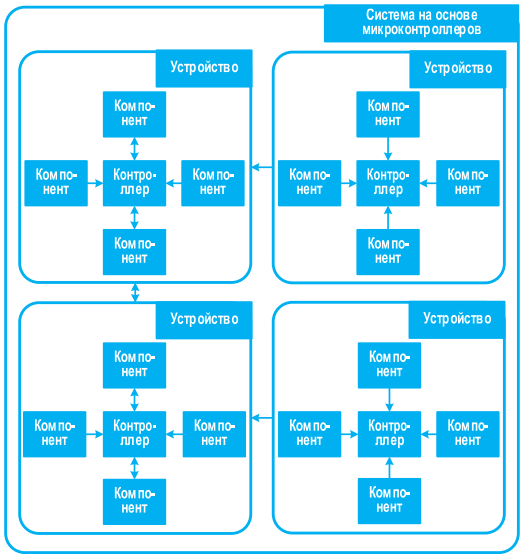

Рис. 1. Представление систем на основе микроконтроллеров

Fig. 1. Architecture of Microcontroller-Based Systems

Рассмотрим каждый тип элемента более подробно, для чего дадим их определения.

Компонент – это то, что возможно подключить к *контроллеру* для отправки данных на него, либо получения данных от него и последующей реакции. *Компоненты* могут обмениваться данными только с теми *контроллерами*, к которым они подключены. *Компоненты* могут быть представлены в виде различных датчиков, приемников, передатчиков, считывателей, двигателей и т. д. Важно отметить, что в данном представлении рассматриваются только готовые компоненты, без учета составляющих электронных схем, например, резисторов, конденсаторов, транзисторов, диодов и др.

Контроллер – это то, что можно запрограммировать для работы с *компонентами* и другими *контроллерами*. *Контроллеры* могут обмениваться данными с *компонентами* и другими *контроллерами*, которые к ним подключены. *Контроллеры* могут быть представлены в виде различных микроконтроллеров и одноплатных компьютеров.

Устройство – это то, что представляет собой совокупность *контроллеров* и *компонентов*, взаимодействующих друг с другом для решения определенных задач. *Устройства* могут взаимодействовать только с другими *устройствами*. *Устройства* могут быть представлены в виде серверов, роботов, станций, дронов, терминалов и т. д.

Подобное представление позволяет в начале проектировать системы на основе микроконтроллеров на уровне устройств, а затем отдельно рассмотреть их внутреннюю структуру. При этом внутри каждого устройства возможно в начале смоделировать взаимосвязи между контроллерами и только после этого рассматривать связь между контроллерами и компонентами.

Допустимы следующие среды передачи данных: *контроллер* ↔ *компонент*, *контроллер* ↔ *контроллер*, *устройство* ↔ *устройство*, *система* ↔ *система*.

4. Иерархическая модель

Для моделирования систем на основе микроконтроллеров защищенными от киберфизических атак был выбран компонентный подход, как наиболее подходящий для учета требований к защищенности системы на ранних стадиях ее жизненного цикла. Модель, предложенная в данном разделе, представляет системы на основе микроконтроллеров в качестве расширяемой иерархической реляционной структуры на основе множеств. При этом для описания модели был использован теоретико-множественный подход. Рассмотрим данное представление более подробно.

Любая система на основе микроконтроллеров $mbs \in MBS$ может быть представлена как:

$$mbs = (MBS', BB, L_{mbs}, a, AA, p_{mbs}),$$

где MBS' – множество подсистем на основе микроконтроллеров системы mbs; BB – блоков системы; L_{mbs} – связей между BB и MBS' системы; a – злоумышленник, атакующий систему; AA – атакующих действий, которые могут быть направлены на систему; p_{mbs} – свойства системы.

В качестве примера mbs могут быть использованы такие системы, как система контроля и управления доступом, система пожарной и/или охранной сигнализации, система видеонаблюдения, система контроля периметра и т. д. При этом ситуация, когда mbs содержит MBS', относится к интегрированным системам, объединяющим, например, контроль доступа с системами пожарной и охранной сигнализации.

Блок системы mbs может быть представлен следующим образом:

$$bb = (BB', HW, SW, L_{bb}, p_{bb}),$$

где BB' – множество подблоков блока bb; HW – аппаратных элементов блока bb; SW – программных элементов блока bb; L_{bb} – связей между программными и аппаратными элементами блока bb; p_{bb} – свойства блока bb.

В качестве примера bb могут быть использованы контроллеры, а также их комбинации с различными компонентами. Это может быть одноплатный компьютер Raspberry Pi, microSD карта памяти с предустановленной операционной системой, микроконтроллер ESP8266 или Iskra JS, а также полноценные устройства – сервер, хаб, робот, станция, дрон и т. п.

Аппаратный элемент системы mbs может быть представлен следующим образом:

$$hw = (HW', L_{hw}, p_{hw}),$$

где HW' – множество аппаратных подэлементов аппаратного элемента hw; L_{hw} – связей между элементами hw; p_{hw} – свойства аппаратного элемента hw. В качестве примера hw могут быть использованы любые компоненты: датчики, приемники, передатчики, считыватели, моторы, аккумуляторы и т. п., а также их комбинации.

Программный элемент системы mbs может быть представлен следующим образом:

$$sw = (SW', L_{sw}, p_{sw}),$$

где SW' – множество программных подэлементов программного элемента sw; L_{sw} – связей между элементами sw; p_{sw} – свойства программного элемента sw. В качестве примера sw может быть использован любой алгоритм, библиотека, прошивка, база данных, приложение или конфигурация.

Связи между элементами системы mbs могут быть представлены следующим образом:

$$L = (R, I, E, p_L),$$

где R – множество протоколов связи, задействованных в L; I – интерфейсов связи, задействованных в L; E – сторон связи, задействованных в L; p_L – свойства связей L. При этом связи между элементами mbs в рамках модели разделены следующим образом:

L_{mbs} – между устройствами системы;
L_{bb} – между контроллерами устройств;
L_{hw} – между контроллерами и компонентами;
L_{sw} – между программными элементами.

Это указывает на то, что модель позволяет представлять протоколы низкого уровня между контроллерами и компонентами вместе с соеди-

нениями между различными алгоритмами внутри прошивки одного из контроллеров, при этом имея возможность представлять протоколы высокого уровня между устройствами (таблица 2).

ТАБЛИЦА 2. Типы связей между элементами моделей

TABLE 2. Link Types between Model Elements

Тип	*Пример*	*R*	*I*	*E*
L_{mbs}	Wi-Fi	IEEE 800.11	wireless 2,4 GHz	устройство ↔ устройство
	ZigBee	IEEE 802.15.4	wireless 2,4 GHz	
	Bluetooth	IEEE 802.15.1	wireless 2,4 GHz	
	nRF24L01+	ESB	wireless 2,4 GHz	
	Infrared	NEC	wireless 38 kHz	
L_{bb}	I2C	SDA + SCL	TWI	контроллер ↔ контроллер
	Serial	TxRx	UART	
	RS-232	RS232	UART	
	RS-485	RS485	UART	
L_{bb}	pin-to-pin	общее питание	shield	контроллер ↔ компонент
	SVG	AVR I/O pin	трехпроводной	
	VG	AVR I/O pin	трехпроводной	
L_{bb}	метод	функции	компилятор	программный элемент ↔ программный элемент
	база данных	SQL-запросы	psycopg2	
	API	JSON-структуры	POST/GET	

В рамках разработанной модели все элементы связаны друг с другом через свойства. Данные свойства могут быть представлены следующим образом:

$$p = (FR, NL, PF, PR),$$

где FR – множество функциональных требований, выполнение которых необходимо для работы элемента; NL – нефункциональных ограничений, выполнение которых необходимо для работы элемента; PF – предоставляемых функциональных возможностей; PR – предоставляемых ресурсов.

В качестве FR может быть использована любая функциональность, необходимая для работы элемента системы – источник питания, защищенное соединение, протокол, интерфейс, загрузчик, библиотека, операционная система, компилятор, драйвер и т. п.; NL – ограничения, необходимые для его работы – место размещения, условия среды, напряжение, ток, флэш-память, цифровое или аналоговое подключение, место на жестком диске, оперативная память и т. п.; PF – функционал, предоставляемый элементом системы – контроль доступа, мониторинг периметра, навигация, обнаружение препятствий, работа с компонентом, шифрование, аутентификация, обработка сигнальных данных и т. п.; PR – ресурс, предоставляемый элементом системы – хранилище данных, вычислительные ресурсы, среда для запуска приложений, возможность добавления / удаления / замены компонентов, возможность взаимодействия со средой и т. п.

Злоумышленник, от которого необходимо защитить систему mbs, может быть представлен следующим образом:

$$a = (ac, kn, rs),$$

где ac – тип доступа a к mbs (от 1 до 5, таблица 3); kn – тип знаний a о mbs (от 1 до 4, таблица 4); rs – тип ресурсов, доступный a для компрометации mbs (от 1 до 3, таблица 5). При этом в разработанной модели структура типов доступа, знаний и ресурсов злоумышленника носит иерархический характер.

ТАБЛИЦА 3. Возможные типы доступа

TABLE 3. Access Types

№ п/п	Описание
1	Нет доступа к системе
2	Доступ к системе через глобальные сети
3	Доступ к системе через локальные сети
4	Физический доступ к системе
5	Полный доступ к системе

ТАБЛИЦА 4. Возможные типы знаний

TABLE 4. Knowledge Types

№ п/п	Описание
1	Общие знания о системе из публично доступных источников
2	Знания о параметрах системы
3	Знания о средствах защиты системы
4	Знания о программном и аппаратном обеспечении системы

ТАБЛИЦА 5. Возможные типы ресурсов

TABLE 5. Resources Types

№ п/п	Описание
1	Общеизвестные программные инструменты и уязвимости
2	Специализированные инструменты и уязвимости нулевого дня
3	Возможность исследования системы

Атакующее действие, направленное на систему mbs, можно представить следующим образом:

$$aa = (cl, oj, sj),$$

где cl – класс aa; oj – объект aa, связывает aa с элементом или элементами системы mbs; sj – субъект aa, связывает aa с a, характеристики которого достаточны для успешной реализации aa.

Классы атакующих действий могут быть представлены на следующих уровнях:

$$cl = \{cn, cr, dv, st\},$$

где cn – компонентов и их взаимосвязей с контроллерами системы; cr – контроллеров и их взаимосвязей с другими контроллерами системы; dv – устройств и их взаимосвязей с другими устройствами системы; st – системы и их взаимосвязей с другими системами.

В качестве атакующих действий могут быть приведены следующие примеры:

– на уровне cn: $cn = \{gie, bcd, rpt, rmt\}$,

где gie – создание некорректных событий от компонентов системы; bcd – обход алгоритмов обнаружения компонентов системы; rpt – замена компонентов системы; rmt – извлечение компонентов системы;

– на уровне cr: $cr = \{rfw, rbl, mup, imw\}$,

где rfw – замена прошивки контроллера; rbl – переустановка загрузчика (bootloader) контроллера; mup – вмешательство в работу системы обновления контроллера; imw – вмешательство в работу проводных каналов связи контроллера.

– на уровне dv: $dv = \{vau, cad, iec, iws\}$,

где vau – вмешательство в работу системы аутентификации устройства; cad – криптографический анализ передаваемых устройством данных; iec – атака, направленная на повышение энергопотребления устройства; iws – вмешательство в работу беспроводных каналов связи устройства.

– на уровне st: $st = \{soc, pwr, web, dbd\}$,

где soc – применение социальной инженерии для получения доступа к системе; pwr – прекращение энергоснабжения системы; web – вмешательство в работу веб-сервисов системы; dbd – вмешательство в работы баз данных системы.

Отметим, что в данной модели компоненты, обеспечивающие защищенность системы, не выделены в качестве отдельного элемента теоретико-множественного представления. Это связано с тем, что большинство таких компонентов представимы как программных и аппаратных элементов, в то время как остальные – как рекомендаций по работе с системой, ее обслуживанием и поддержкой.

4. Взаимосвязи моделей

Рассмотрим, как классы атакующих действий (cl) связаны с параметрами злоумышленников (ac, kn, rs), необходимыми для их реализации (таблица 6). В качестве примера рассмотрим ситуацию, когда проектируемая mbs должна быть защищена от a с параметрами $ac = 4$, $kn = 2$ и $rs = 2$. Выделение границ ячеек таблицы толщиной линий и цветом показывает значения параметров злоумышленника. Связи между возможностью реализации атакующих действий и достаточными значениями параметров злоумышленника показаны знаком «+». В соответствии с содержанием таблицы, проектируемая mbs должна быть защищена от rpt, rmt, imw, iec, iws, soc, pwr, web и dbd. Данные атакующие действия также выделены в таблице.

ТАБЛИЦА 6. Связи между атакующими действиями и параметрами злоумышленника

TABLE 6. Connections between Attack Actions and Attacker Parameters

Атакующие действия			a											
			ac					kn				rs		
			1	2	3	4	5	1	**2**	3	4	1	2	3
cl	cn	gie			+	+	+		+	+	+			+
		bcd			+	+	+			+	+		+	+
		rpt				+	+		+	+	+	+	+	+
		rmt				+	+	+	+	+	+	+	+	+
	cr	rfw				+	+				+			+
		rbl				+	+				+			+
		mup			+	+	+			+	+		+	+
		imw				+	+	+	+	+	+	+	+	+
	dv	vau			+	+	+			+	+		+	+
		cad			+	+	+			+	+		+	+
		iec			+	+	+		+	+	+	+	+	+
		iws			+	+	+	+	+	+	+		+	+
	st	***soc***	+	+	+	+	+	+	+	+	+	+	+	+
		pwr				+	+	+	+	+	+	+	+	+
		web		+	+	+	+		+	+	+	+	+	+
		dbd		+	+	+	+		+	+	+	+	+	+

Как было упомянуто ранее, в разработанной модели структура параметров злоумышленника иерархична. Иерархичность параметров влияет на то, что злоумышленник с определенным типом доступа к системе имеет возможность осуществить любое атакующее действие, которое возможно для злоумышленников с таким же типом доступа, но с меньшими знаниями/ресурсами. Такая зависимость позволяет хранить данные только о пороговых значениях параметров, необходимых для успешной реализации атакующих действий. При этом разработанная модель позволяет использо-

вать различные модели атакующих и атакующих действий, изменяя количество параметров и диапазоны их значений. Точно так же для атакующих действий можно использовать другую классификацию, а примеры можно расширить. Также рассмотрим, как классы атакующих действий связаны с элементами защиты систем на основе микроконтроллеров (таблица 7).

ТАБЛИЦА 7. Связи между атакующими действиями и элементами защиты

TABLE 7. Connections between Attack Actions and Security Elements

Атакующие действия			Элементы защиты
cl	*cn*	*gie*	алгоритм обнаружения аномалий, скрытое размещение датчиков
		bcd	алгоритм корреляции событий, скрытое размещение датчиков
		rpt	антивандальный корпус, аппаратная аутентификация
		rmt	антивандальный корпус
	cr	*rfw*	антивандальный корпус, шифрование прошивки
		rbl	антивандальный корпус, шифрование загрузчика
		mup	антивандальный корпус, удаление физического интерфейса обновления
		imw	антивандальный корпус, шифрование, аутентификация
	dv	*vau*	надежные учетные данные для входа, политика паролей, защита от грубой силы
		cad	надежные алгоритмы шифрования, безопасный механизм распределения ключей
		iec	обнаружение аномалий на основе поведения, изоляция/ограничение устройств
		iws	надежный алгоритм шифрования в точке доступа, надежные учетные данные для входа в систему, аутентификация на основе пары открытых ключей
	st	*soc*	обучение операторов и пользователей, политика безопасности
		pwr	источники бесперебойного питания, резервный источник питания
		web	брандмауэр, механизм обновления, механизм резервного копирования, механизм ведения журнала
		dbd	проверка ввода, строгая политика доступа, надежные учетные данные для входа, отдельные пользователи базы данных для разных операций

В разработанной модели возможные атакующие действия определяются элементами системы, а также параметрами злоумышленника, от которого данную систему необходимо защитить. При этом для предотвращения возможности реализации атакующих действий могут быть использованы различные элементы защиты, интерпретация которых в качестве программных (например, алгоритм обнаружения аномалий) и аппаратных (антивандальный корпус) элементов, а также рекомендаций (обучение операторов и пользователей) позволяет в рамках разработанной модели сделать их встраивание неотъемлемой частью процесса проектирования защищенных систем.

Рассмотрим, как возможность реализации классов атакующих действий зависит от наличия в системе элементов, не относящихся к элементам защиты (таблица 8). Важно отметить, что взаимосвязи таких элементов с возможностью реализации атакующих действий, дополненные информацией о параметрах злоумышленника, определяют совокупность атакующих действий, защиту от которых необходимо обеспечить.

Данные о релевантных атакующих действиях позволяют проработать защиту от них еще на ранних стадиях жизненного цикла системы, что повышает не только текущий уровень ее безопасности, но и облегчает дальнейший процесс поддержки уровня защищенности при эксплуатации системы.

5. Область применения

Результаты сравнения разработанной модели с другими решениями сведены в единую таблицу 9, где использованы следующие условные обозначения: «+» – поддержка анализируемого параметра; «–» – отсутствие поддержки; «*» – потенциальная возможность поддержки данного параметра. Сравнение было выполнено по уровням систем на основе микроконтроллеров, описание которых возможно с помощью проанализированных моделей, а также классам атак, учет которых возможен в рамках данных моделей.

При этом выделены следующие уровни системы:

– $cn \leftrightarrow cr$, т. е. контроллеры, компоненты и их взаимодействия;

– $cr \leftrightarrow cr$, т. е. контроллеры и их взаимодействие друг с другом;

– $dv \leftrightarrow dv$, т. е. устройства и их взаимодействие друг с другом;

– $st \leftrightarrow st$, т. е. система и ее взаимодействие с другими системами.

В свою очередь, атакующие действия (атаки) разделены на следующие классы:

– *cn* (на компоненты и их связь с контроллерами);

– *cr* (на контроллеры и их связь с другими контроллерами);

– *dv* (на устройства и их связь с другими устройствами);

– *st* (на систему и ее связь с другими системами).

ТАБЛИЦА 8. Связи между атакующими действиями и элементами систем

TABLE 8. Connections between Attack Actions and System Elements

Атакующие действия			*mbs*
cl	*cn*	*gie*	компоненты, измеряющие параметры окружающей среды
		bcd	компоненты, контролирующие изменение параметров окружающей среды
		rpt	любой компонент
		rmt	любой компонент
	cr	*rfw*	любой контроллер с возможностью перезаписи прошивки
		rbl	любой контроллер с возможностью перезаписи загрузчика
		mup	любой контроллер с системой обновления
		imw	контроллер ↔ контроллер, контроллер ↔ компонент
	dv	*vau*	устройство ↔ устройство, если в процессе взаимодействия используется аутентификация
		cad	устройство ↔ устройство, если в процессе взаимодействия используется шифрование
		iec	любое устройство с возможностью гибернации и/или беспроводными интерфейсами
		iws	устройство ↔ устройство
	st	*soc*	любая система с пользователями и/или операторами
		pwr	любая система с питанием по сети
		web	любая система с веб-сервисами
		dbd	любая система с базой данных

ТАБЛИЦА 9. Сравнение с другими решениями

TABLE 9. Comparison with Other Solutions

Решения	Уровни системы				Классы атак			
	cn ↔ *cr*	*cr* ↔ *cr*	*dv* ↔ *dv*	*st* ↔ *st*	*cn*	*cr*	*dv*	*st*
[17]	–	–	–	–	–	*	*	–
[18]	–	–	–	–	–	*	*	–
[19]	+	+	–	–	+	+	–	–
[21]	+	+	–	–	–	+	–	–
[23]	+	+	–	–	+	+	–	–
[26]	–	–	+	–	–	+	+	–
Предлагаемое	+	+	+	+	+	+	+	+

Сравнение показало, что разработанная модель обеспечивает учет всех анализируемых классов атакующих действий, а также позволяет описать проектируемую систему на основе микроконтроллеров на всех представленных уровнях, в то время как другие решения уступают в данных параметрах.

6. Заключение

В данной работе предложена иерархическая модель для проектирования систем на основе микроконтроллеров защищенными от киберфизических атак. Описание модели выполнено в рамках теоретико-множественного подхода.

При этом были представлены следующие элементы модели: система в целом, программно-аппаратные блоки, программные и аппаратные элементы, связи между элементами (на уровне устройств, контроллеров, компонентов и программного обеспечения), свойства элементов (предоставляемый и запрашиваемый функционал/ресурсы), злоумышленник, атакующие действия, а также их классификация. Были показаны взаимосвязи между элементами модели: параметры злоумышленника, достаточные для успешной реализации атакующих действий; элементы систем на основе микроконтроллеров, необходимые для реализации атакующих действий; элементы защиты, достаточные для предотвращения атакующих действий.

В рамках дальнейших исследований планируется расширение предлагаемой модели инструментами имитационного моделирования для динамического анализа защищенности систем. Также планируется апробация предложенной модели в различных приложениях.

Список источников

1. Левшун Д.С., Гайфулина Д.А., Чечулин А.А., Котенко И.В. Проблемные вопросы информационной безопасности киберфизических систем // Информатика и автоматизация. 2020. Т. 19. № 5. С. 1050–1088. DOI:10.15622/ia.2020.19.5.6

2. Левшун Д.С., Чечулин А.А., Котенко И.В. Жизненный цикл разработки защищенных систем на основе встроенных устройств // Защита информации. Инсайд. 2017. № 4. С. 53–59.

3. Котенко И.В., Чечулин А.А., Левшун Д.С. Анализ защищенности инфраструктуры железнодорожного транспорта на основе аналитического моделирования // Защита информации. Инсайд. 2017. № 6. С. 48–57.

4. Левшун Д.С., Чечулин А.А., Котенко И.В. Комплексная модель защищенных киберфизических систем для их проектирования и верификации // Труды учебных заведений связи. 2019. Т. 5. № 4. С. 114–123. DOI:10.31854/1813-324X-2019-5-4-114-123

5. Левшун Д.С., Чечулин А.А., Котенко И.В. Проектирование безопасной среды передачи данных на примере протокола I2C // Защита информации. Инсайд. 2018. № 4. С. 54–62.

6. Islam J., Habiba U., Kabir H., Martuza KG., Akter F., Hafiz F., et al. Design and Development of Microcontroller Based Wireless Humidity Monitor // IOSR Journal of Electrical and Electronics Engineering // 2018. Vol. 13. Iss. 2. PP. 41–46. DOI:10.9790/1676-1302034146

7. Sowah RA., Boahene DE., Owoh DC., Addo R., Mills G.A., Owusu-Banahene W., et al. Design of a Secure Wireless Home Automation System with an Open Home Automation Bus (OpenHAB 2) Framework // Journal of Sensors. 2020. Vol. 2020. PP. 1–22. DOI:10.1155/2020/8868602

8. Tebueva F.B., Rosenko A.P., Nechvoloda V.E. Development of Methods and Software Modules Security Assessment Information of Limited Distribution // Proceedings of the Young Scientist's Third International Workshop on Trends in Information Processing (YSIP3 2019, Stavropol, Russia, 17–20th September 2019). CEUR, 2019.

9. Mutharasu S., Divya V., Bharathi D.K., Elakkiya M.V., Janani E. Design and implementation of agrobot by using IoT // International Journal of Advance Research, Ideas and Innovations in Technology. 2019. Vol. 5. Iss. 2. PP. 10–13.

10. Kovtunenko A., Bilyalov A., Valeev S. Distributed Streaming Data Processing in IoT Systems Using Multi-agent Software Architecture // Proceedings of the 18th International Conference on Next Generation Wired/Wireless Networking, NEW2AN 2018, and 11th Conference on Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ruSMART 2018, (St. Petersburg, Russia, 27–29 August 2018). Lecture Notes in Computer Science. Vol. 11118. Cham: Springer, 2018. PP. 572–583. DOI:10.1007/978-3-030-01168-0_51

11. Ceken C., Abdurahman D. Simulation Modeling of An IoT Based Cold Chain Logistics Management System // Sakarya University Journal of Computer and Information Sciences. 2019. Vol. 2. Iss. 2. PP. 89–100. DOI:10.35377/saucis.02.02.598963

12. Ashouri M., Lorig F., Davidsson P., Spalazzese R. Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics // Future Internet. 2019. Vol. 11. Iss. 11. P. 235. DOI:10.3390/fi11110235

13. Giménez P., Molína B. Palau C.E., Esteve M. SWE Simulation and Testing for the IoT // Proceedings of the International Conference on Systems, Man, and Cybernetics (Manchester, UK, 13–16 October 2013). IEEE, 2013. PP. 356–361. DOI:10.1109/SMC.2013.67

14. Andres-Maldonado P., Lauridsen M., Ameigeiras P., Lopez-Soler J.M. Analytical Modeling and Experimental Validation of NB-IoT Device Energy Consumption // IEEE Internet of Things Journal. 2019. Vol. 6. Iss. 3. PP. 5691–5701. DOI:10.1109/JIOT.2019.2904802

15. Udoh S.J., Srivastava V.M. Analytical Modeling of Radio Network Performance for 5G (Non-Standalone) and It's Network Connectivity // Journal of Communications. 2020. Vol. 15. Iss. 12. PP. 886–895. DOI:10.12720/jcm.15.12.886-895

16. Bhavana A., Nandha Kumar A.N. An Analytical Modeling for Leveraging Scalable Communication in IoT for Inter-Domain Routing // Proceedings of the Computational Methods in Systems and Software (CoMeSySo 2018, 12–14 September 2018). Advances in Intelligent Systems and Computing. Vol. 859. Cham: Springer, 2018. PP. 1–11. DOI:10.1007/978-3-030-00211-4_1

17. Hu F., Lu Y., Vasilakos A.V., Hao Q., Ma R., Patil Y., et al. Robust Cyber–Physical Systems: Concept, models, and implementation // Future Generation Computer Systems. 2016. Vol. 56. PP. 449–475. DOI:10.1016/j.future.2015.06.006

18. Penas O., Plateaux R., Patalano S., Hammadi M. Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems // Computers in Industry. 2017. Vol. 86. PP. 52–69. DOI:10.1016/j.compind.2016.12.001

19. Scott-Hayward S. Design and deployment of secure, robust, and resilient SDN controllers // Proceedings of the 1st Conference on Network Softwarization (London, UK, 13–17 April 2015). IEEE, 2015. PP. 1–5. DOI:10.1109/NETSOFT.2015.7258233

20. Hu F., Hao Q., Bao K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation // IEEE Communications Surveys & Tutorials. 2014. Vol. 16. Iss. 4. PP. 2181–2206. DOI:10.1109/COMST.2014.2326417

21. Lin Z., Yu S., Lü J., Cai S., Chen G. Design and ARM-Embedded Implementation of a Chaotic Map-Based Real-Time Secure Video Communication System // IEEE Transactions on Circuits and Systems for Video Technology. 2014. Vol. 25. Iss. 7. PP. 1203–1216. DOI:10.1109/TCSVT.2014.2369711

22. National Institute of Standards and Technology // NIST. URL: https://www.nist.gov (дата обращения 3.02.2023)

23. Desnitsky V., Kotenko I., Chechulin A. Configuration-based approach to embedded device security // Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS 2012, St. Petersburg, Russia, 17–19 October 2012). Lecture Notes in Computer Science. Vol. 7531. Berlin, Heidelberg: Springer, 2012. PP. 270–285. DOI:10.1007/978-3-642-33704-8_23

24. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems. 2011. URL: https://www.omg.org/spec/MARTE/1.1/PDF (дата обращения 3.02.2023)

25. Chechulin A., Kotenko I., Desnitsky V. An Approach for Network Information Flow Analysis for Systems of Embedded Components // Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS 2012, St. Petersburg, Russia, 17–19 October 2012). Lecture Notes in Computer Science. Vol. 7531. Berlin, Heidelberg: Springer, 2012. PP. 146–155. DOI:10.1007/978-3-642-33704-8_13

26. Design of Secure and Energy-Efficient Embedded Systems for Future Internet Applications // CORDIS. URL: https://cordis.europa.eu/project/id/256668 (дата обращения 3.02.2023)

References

1. Levshun D., Gaifulina D., Chechulin A., Kotenko I. Problematic Issues of Information Security of Cyber-Physical Systems. *Informatics and Automation*. 2020;19(5):1050–1088. (in Russ.) DOI:10.15622/ia.2020.19.5.6

2. Levshun D., Chechulin A., Kotenko I. Design lifecycle for secure cyber-physical systems based on embedded devices. *Zašita informacii. Inside*. 2017;4(76):53–59. (in Russ.)

3. Kotenko I., Chechulin A., Levshun D. Security analysis of railway transport infrastructure on the base of analytical modeling. *Zaŝita informacii. Inside*. 2017;6(78):48–57. (in Russ.)

4. Levshun D., Chechulin A., Kotenko I. A Comprehensive Model of Secure Cyber-Physical Systems for their Design and Verification. *Proc. of Telecom. Universities*. 2019;5(4):114–123 (in Russ.) DOI:10.31854/1813-324X-2019-5-4-114-123

5. Levshun D., Chechulin A., Kotenko I. Design of Secure Data Transfer Environment Using the I2C Protocol as an Example. *Zašita informacii. Inside*. 2018; 4(82):54–62. (in Russ.)

6. Islam J., Habiba U., Kabir H., Martuza KG., Akter F., Hafiz F., et al. Design and Development of Microcontroller Based Wireless Humidity Monitor. *IOSR Journal of Electrical and Electronics Engineering.* 2018;13(2):41–46. DOI:10.9790/1676-1302034146

7. Sowah RA., Boahene DE., Owoh DC., Addo R., Mills G.A., Owusu-Banahene W., et al. Design of a Secure Wireless Home Automation System with an Open Home Automation Bus (OpenHAB 2) Framework. *Journal of Sensors*. 2020;2020:1–22. DOI:10.1155/2020/8868602

8. Tebueva F.B., Rosenko A.P., Nechvoloda V.E. Development of Methods and Software Modules Security Assessment Infor-mation of Limited Distribution. *Proceedings of the Young Scientist's Third International Workshop on Trends in Information Processing, YSIP3 2019, 17–20th September 2019, Stavropol, Russia.* CEUR; 2019.

9. Mutharasu S., Divya V., Bharathi D.K., Elakkiya M.V., Janani E. Design and implementation of agrobot by using IoT. *International Journal of Advance Research, Ideas and Innovations in Technology*. 2019;5(2):10–13.

10. Kovtunenko A., Bilyalov A., Valeev S. Distributed Streaming Data Processing in IoT Systems Using Multi-agent Software Architecture. *Proceedings of the 18th International Conference on Next Generation Wired/Wireless Networking, NEW2AN 2018, and 11th Conference on Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ruSMART 2018, 27–29 August 2018, St. Petersburg, Russia. Lecture Notes in Computer Science. vol.11118.* Cham: Springer; 2018. p.572–583. DOI:10.1007/978-3-030-01168-0_51

11. Ceken C., Abdurahman D. Simulation Modeling of An IoT Based Cold Chain Logistics Management System. *Sakarya University Journal of Computer and Information Sciences*. 2019;2(2):89–100. DOI:10.35377/saucis.02.02.598963

12. Ashouri M., Lorig F., Davidsson P., Spalazzese R. Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics. *Future Internet*. 2019;11(11):235. DOI:10.3390/fi11110235

13. Giménez P., Molína B. Palau C.E., Esteve M. SWE Simulation and Testing for the IoT. *Proceedings of the International Conference on Systems, Man, and Cybernetics, 13–16 October 2013, Manchester, UK*. IEEE; 2013. PP. 356–361. DOI:10.1109/SMC.2013.67

14. Andres-Maldonado P., Lauridsen M., Ameigeiras P., Lopez-Soler J.M. Analytical Modeling and Experimental Validation of NB-IoT Device Energy Consumption. *IEEE Internet of Things Journal*. 2019;6(3):5691–5701. DOI:10.1109/JIOT.2019.2904802

15. Udoh S.J., Srivastava V.M. Analytical Modeling of Radio Network Performance for 5G (Non-Standalone) and It's Network Connectivity. *Journal of Communications*. 2020;15(12):886–895. DOI:10.12720/jcm.15.12.886-895

16. Bhavana A., Nandha Kumar A.N. An Analytical Modeling for Leveraging Scalable Communication in IoT for Inter-Domain Routing. *Proceedings of the Computational Methods in Systems and Software, CoMeSySo 2018, 12–14 September 2018. Advances in Intelligent Systems and Computing. vol.859*. Cham: Springer; 2018. p.1–11. DOI:10.1007/978-3-030-00211-4_1

17. Hu F., Lu Y., Vasilakos A.V., Hao Q., Ma R., Patil Y., et al. Robust Cyber–Physical Systems: Concept, models, and implementation. *Future Generation Computer Systems*. 2016;56:449–475. DOI:10.1016/j.future.2015.06.006

18. Penas O., Plateaux R., Patalano S., Hammadi M. Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems. *Computers in Industry*. 2017;86:52–69. DOI:10.1016/j.compind.2016.12.001

19. Scott-Hayward S. Design and deployment of secure, robust, and resilient SDN controllers. *Proceedings of the 1st Conference on Network Softwarization, NetSoft, 13–17 April 2015, London, UK.* IEEE; 2015. p.1–5. DOI:10.1109/NETSOFT.2015.7258233

20. Hu F., Hao Q., Bao K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. *IEEE Communications Surveys & Tutorials*. 2014;16(4):2181–2206. DOI:10.1109/COMST.2014.2326417

21. Lin Z., Yu S., Lü J., Cai S., Chen G. Design and ARM-Embedded Implementation of a Chaotic Map-Based Real-Time Secure Video Communication System. *IEEE Transactions on Circuits and Systems for Video Technology*. 2014;25(7):1203–1216. DOI:10.1109/TCSVT.2014.2369711

22. *NIST*. National Institute of Standards and Technology. URL: https://www.nist.gov [Accessed 3rd February 2023]

23. Desnitsky V., Kotenko I., Chechulin A. Configuration-based approach to embedded device security. *Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012, 17–19 October 2012, St. Petersburg, Russia. Lecture Notes in Computer Science. vol.7531.* Berlin, Heidelberg: Springer; 2012. p.270–285. DOI:10.1007/978-3-642-33704-8_23

24. *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems*. 2011. URL: https://www.omg.org/spec/MARTE/1.1/PDF [Accessed 3rd February 2023]

25. Chechulin A., Kotenko I., Desnitsky V. An Approach for Network Information Flow Analysis for Systems of Embedded Components. *Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012, 17–19 October 2012, St. Petersburg, Russia. Lecture Notes in Computer Science. vol.7531*. Berlin, Heidelberg: Springer; 2012. p.146–155. DOI:10.1007/978-3-642-33704-8_13

26. *CORDIS*. Design of Secure and Energy-Efficient Embedded Systems for Future Internet Applications. URL: https://cordis.europa. eu/project/id/256668 [Accessed 3rd February 2023]

Статья поступила в редакцию 06.02.2023; одобрена после рецензирования 14.02.2023; принята к публикации 15.02.2023.
The article was submitted 06.02.2023; approved after reviewing 14.02.2023; accepted for publication 15.02.2023.

Информация об авторе:

ЛЕВШУН Дмитрий Сергеевич | старший научный сотрудник Санкт-Петербургского Федерального исследовательского центра Российской академии наук,
https://orcid.org/0000-0003-1898-6624