# КОМПЛЕКСНАЯ МОДЕЛЬ ЗАЩИЩЕННЫХ КИБЕРФИЗИЧЕСКИХ СИСТЕМ ДЛЯ ИХ ПРОЕКТИРОВАНИЯ И ВЕРИФИКАЦИИ

**Д.С. Левшун**[1, 2*], **А.А. Чечулин**[1, 3], **И.В. Котенко**[1, 3]

[1]Санкт-Петербургский институт информатики и автоматизации Российской академии наук,
Санкт-Петербург, 199178, Российская Федерация
[2]Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики,
Санкт-Петербург, 197101, Российская Федерация
[3]Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация
*Адрес для переписки: levshun@comsec.spb.ru

**Информация о статье**
УДК 004.056.53
Статья поступила в редакцию 05.11.2019

**Ссылка для цитирования:** 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

**Аннотация:** *В статье предложена комплексная модель защищенных киберфизических систем, предназначенная для их проектирования и верификации. В рамках данной модели киберфизическая система представляется в виде множества блоков с различными свойствами и связями между ними. Основная сложность при построении подобной модели заключается в объединении различных подходов к моделированию киберфизических систем (или их элементов) в рамках единого подхода. Основная цель предлагаемого подхода к моделированию заключается в обеспечении возможности преобразования различных моделей друг в друга без потери значимых данных о блоках системы, а также учете эмерджентных свойств, возникающих в процессе их взаимодействия. Корректность предложенной модели обоснована на примере ее использования для проектирования и верификации системы контроля и управления доступом.*

**Ключевые слова:** *безопасность на основе проектирования, киберфизическая система, верификация безопасности, моделирование систем, модель атакующего, модель атакующих действий, комплексная модель.*

## 1. Введение

Киберфизические системы (КФС) представляют собой системы со сложной структурой, которые объединяют множество различных физических и программных элементов. КФС могут быть распределенными, децентрализованными и самоорганизующимися, а также содержать множество устройств на основе микроконтроллеров [1]. Как следствие, существуют различные подходы для проектирования подобных систем как защищенных систем. При этом некоторые ориентированы на программные или аппаратные элементы систем, в то время как другие применимы только в рамках конкретной области приложения: автомобильная инфраструктура, роботы, умные дома, индустриальный интернет вещей и т. д. Ключевой особенностью каждого из подходов является используемая в нем модель КФС и набор свойств ее элементов.

С помощью разработанных инструментов и подходов могут быть смоделированы различные аспекты КФС: физические процессы, производительность, баланс нагрузки, поведение, топология, иерархия, бизнес процессы, отдельные элементы и вычислительные платформы, сети и среды передачи данных, многое другое. При этом большинство подходов ориентировано на обеспечение стабильности и надежности систем, а не на обеспечение их защищенности. И хотя для отдельных элементов КФС существуют решения для обеспечения безопасности на основе проектирования и верификации, аналогичного решения для КФС в целом на данный момент в научной литературе не представлено. Более того, преобразование одной из существующих моделей в другую без потери значимых данных является сложной или даже невыполнимой задачей. Именно поэтому важно, чтобы новый подход был основан на комплексной модели, позволяющей объединить модели отдельных аспектов КФС между собой и обеспечить их взаимное преобразование.

Научным вкладом данной работы является новый подход к моделированию защищенных КФС при их проектировании и верификации. В рамках данного подхода используется комплексная модель, позволяющая представить КФС в виде множества взаимодействующих блоков с различными свойствами. Основная сложность при построении подобной модели заключается в объединении различных подходов к моделированию в рамках единого подхода с учетом того, что для различных блоков КФС эффективное представление достигается за счет применения разных моделей. Основной целью предлагаемого подхода является обеспечение взаимного преобразования используемых моделей без потери данных о свойствах блоков КФС, а также с учетом влияния эмерджентных свойств (возникают при взаимодействии блоков КФС). Новизна предлагаемого подхода заключается в интеграции нескольких подходов к моделированию: на основе событий (на уровне среды передачи данных), с помощью направленных графов (на уровне топологии), многоагентное моделирование (на уровне сети передачи данных), моделирование непрерывных процессов (на уровне аппаратных элементов), моделирование дискретных процессов (на уровне программных элементов), моделирование взаимодействия (на уровне блоков) и моделирование на основе онтологий (на уровне КФС, атакующего и атакующих действий). Предлагаемое решение ориентировано на безопасность КФС, что позволяет использовать модели атакующего и атакующих действий в качестве неотъемлемой части комплексной модели. При этом влияние атакующих действий на блоки КФС моделируется через изменения их свойств, в то время как свойства атакующего ограничивают спектр возможных атакующих действий на них.

Статья организована следующим образом. В разделе 2 проанализированы существующие решения в области моделирования КФС для их проектирования и верификации. В разделе 3 предложена новая интегрированная модель защищенной КФС. В разделе 4 представлен пример использования предложенной модели для проектирования и верификации системы контроля и управления доступом. В разделе 5 представлены достоинства, недостатки и область применения предложенной модели. В разделе 6 содержатся основные выводы и направления дальнейших исследований.

## 2. Анализ существующих решений

В работе [2] авторы предложили классификацию подходов к моделированию КФС на основе отражаемых моделью аспектов системы, а также решаемых ею задач. Предложенная классификация выглядит следующим образом:

– модели на основе акторов для задач, связанных с временными параметрами и производительностью системы [3];

– модели, основанные на событиях, для задач, связанных с вычислительными процессами, взаимодействием элементов и управлением ими;

– SCADA-модели (*от англ.* Supervisory Control and Data Acquisition – диспетчерское управление и сбор данных) для задач, связанных с балансировкой нагрузки, проверкой стабильности и целостности [4];

– объединение систем дифференциальных уравнений и конечных автоматов для задач, связанных с работой несложных систем [5];

– непрерывные модели для задач, связанных с физическими процессами [6];

– MDD (*от англ.* Model-Driven Development – разработка, управляемая моделями) [7], MIC (*от англ.* Model-Integrated Computing – интегрированная модель вычислений) [8] и DSM (*от англ.* Domain Specific Modeling – предметно-ориентированное моделирование) [9] для задач, связанных с программными элементами;

– многоагентное моделирование для задач, связанных с взаимодействием элементов системы.

В работе [10] авторы представили математическую модель КФС. Модель была получена с помощью инструмента Modelica [11]. В этой модели КФС была представлена как набор компонентов и связанных с ними интерфейсов. Для отображения динамики физических компонентов авторы использовали непрерывную модель, а для отображения вычислительных компонентов – дискретную модель. Основная сложность представленного исследования заключалась в решении задачи объединения непрерывной и дискретной моделей для определения значимых функциональных и системных параметров с целью дальнейшей оптимизации.

В работе [12] предложена методика проектирования КФС. Рассмотрим 7 основных шагов данной методики более подробно.

*Шаг 1*. Определение границ системы на основе методов черного и белого ящика. Реализация основана на SYSML (*от англ.* The Systems Modeling Language – предметно-ориентированный язык моделирования систем) [13] диаграммах и моделях Dymola [14] / Modelica.

*Шаг 2*. Получение многоуровневого представления на основе подхода MBSE (*от англ.* Model-Based Systems Engineering – системная инженерия на основе моделей или моделе-ориентированная системная инженерия) [15]. Реализация базируется на SYSML-диаграммах и OOM (*от англ.* Object-Oriented Modeling – объектно-ориентированное моделирование, проектирование) [16] для описания спецификаций, анализа, проектирования, верификации и валидации КФС.

*Шаг 3*. Моделирование взаимодействия элементов системы с помощью моделирования на основе портов. Реализация основана на SYSML-диаграммах и моделях Dymola / Modelica.

*Шаг 4*. Моделирование топологии с учетом того, что существующие элементы могут быть представлены в виде множества связанных между собой топологических элементов с подмножествами. Реализация основана на множестве направленных графов, представляющих зависимости между подсистемами, компонентами и связанными с ними параметрами.

*Шаг 5*. Определение семантической совместимости на базе определения существующих точек зрения с помощью онтологической базы знаний и декомпозиции каждой точки зрения с помощью топологического анализа на основе графов. Реализация основана на онтологиях, поддерживающих разметку на основе графов.

*Шаг 6*. Многоагентное моделирование для представления протоколов управления и взаимодействия (задержки связи, взаимодействия, изменения топологии, узлы связи и соединения между ними, потеря пакетов). Реализация основана на топологических графах и многоагентном моделировании.

*Шаг 7*. Моделирование совместной работы для решения проблемы многочисленных точек зрения и проблемы взаимодействия между агентами различных онтологий. Реализация основана на OWL (от англ. Web Ontology Language – язык онтологий веб-документов и приложений) [17], предоставляющем интерфейсы.

Модель, предложенная в работе [12], позволяет отразить следующие аспекты КФС: внешние и внутренние взаимодействия, управление процессом, имитацию поведения, отображение топологических взаимоотношений и совместимость элементов на основе многоагентной платформы.

В работе [18] представлен подход к верификации временных параметров КФС, а также ее производительности. Авторами были смоделированы следующие аспекты КФС: функциональные взаимосвязи между входами и выходами системы; временные параметры элементов системы; взаимодействия между элементами системы; а также ограничения, связанные с синхронизацией элементов системы. Валидация полученной модели была выполнена в среде TrueTime (Matlab/Simulink) [19], а верификация – на основе проверки на модели в среде UPPAAL [20], являющейся инструментарием для верификации и валидации систем реального времени, моделируемых как сети временных автоматов. С помощью проверки на модели авторы подтвердили стабильность, надежность и возможность реализации КФС.

В работе [21] рассмотрен объектно-ориентированный язык для формализации процессов КФС в контексте гетерогенных и динамических сред на основе компонентной метамодели. В рамках данного языка процессы представляют собой модели поведения системы на различных уровнях абстракции, а именно:

– мета-мета модель процесса – представляет собой семантическое и синтаксическое описание элементов и структур;

– мета модель процесса – определяет все элементы, их типы, взаимоотношения и структурные комбинации;

– модель процесса – задает абстрактное описание процесса;

– экземпляр процесса – представляет собой описание конкретного процесса в момент его конкретной реализации.

Процесс представлен в виде структуры, состоящей из следующих элементов: шагов процесса, переходов, данных, событий, логических шагов, процесса и обрабатываемых объектов. Реализация основана на EMF (*от англ*. Eclipse Modeling Framework – фреймворк, смоделированный в Eclipse – свободной интегрированной среде разработки модульных кроссплатформенных приложений) [22].

В работе [23] специфицирован подход к верификации временных параметров систем автоматизации сети – NAS (*от англ.* Network Automation Systems). Предложенный авторами подход состоит из трех основных фаз:

1) построение модели, связанной с определением времени реакции компонентов и измерением их временных характеристик;

2) моделирование, связанное применением компонентных математических моделей для отображения сетевой архитектуры и взаимосвязей между элементами;

3) верификация, связанная с абстрактным представлением NAS в виде формальных моделей на основе конечных автоматов в среде UPPAAL.

На завершающем шаге подхода, заданного в [23], полученная формальная модель системы используется для верификации времени отклика NAS с использованием TCTL [24].

### 3. Комплексная модель

Анализ современного состояния исследований показал, что большинство подходов к моделированию КФС ориентированы на обеспечение их стабильности и надежности, а не на обеспечение защищенности. И хотя некоторые подходы направлены на проектирование отдельных элементов систем с учетом защищенности, требуются комплексные решения по проектированию и верификации защищенных КФС.

В предлагаемой методике проектирования и верификации защищенных КФС модель системы состоит из следующих элементов: блоков (программные и аппаратные элементы), соединений между ними (топология, среда передачи данных), атакующего и атакующих действий.

Абстрактное представление этой модели представлено на рисунке 1. Скругленные закрашенные прямоугольники отражают модель КФС вместе с ее элементами, в то время как черные стрелки отражают их иерархию и вложенность. Скругленные бесцветные прямоугольники отражают внешние модели, которые связаны с комплексной моделью и интегрированы в нее (красные линии).

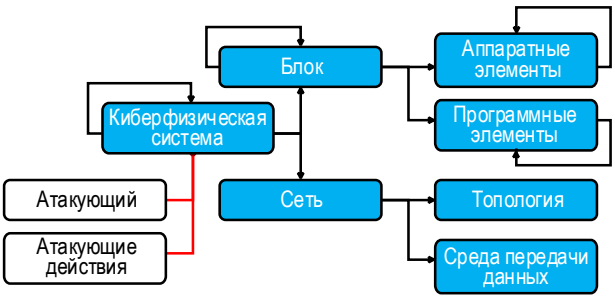


**Рис. 1. Комплексная модель защищенной КФС**

Комплексная модель КФС может быть представлена следующим образом:

$$cps = (CPS^*, BB, nw, a, AA, P_{cps}), \quad (1)$$

где $CPS^*$ – множество киберфизических подсистем $cps^*$ системы $cps$; $BB$ – множество блоков $cps$; $nw$ – сеть передачи данных между блоками $cps$; $a$ – атакующий, заинтересованный в нарушении безопасности $cps$; $AA$ – множество атакующих действий, которые потенциально могут нарушить безопасность $cps$; $P_{cps}$ – множество свойств $cps$.

Отметим, что каждый элемент $cps$ на данном уровне представления рассматривается в качестве объекта с определенными свойствами. При этом внутренняя структура элемента не учитывается (данное правило работает для каждого уровня абстракции по отдельности). Реализация модели $cps$ основана на OWL.

Атакующий, заинтересованный в нарушении безопасности $cps$, может быть представлен следующим образом:

$$a = (tp, lvl, ap), \quad (2)$$

где $tp$ – тип доступа $a$; $lvl$ – уровень возможностей $a$; $ap$ – точка доступа $a$.

Реализация модели $a$ основана на OWL.

Множество атакующих действий, которые потенциально могут нарушить безопасность $cps$, могут быть представлены следующим образом:

$$AA = \{(aa_1, prcnd_1, P_{aa_1}), \dots, (aa_n, prcnd_n, P_{aa_n})\}, \quad (3)$$

где $aa_i \mid i \in 1 \dots n$ – $i$-ое атакующее действие из множества $AA$ со свойствами $P_{aa_i}$; $prcnd_i \mid i \in 1 \dots n$ – $i$-ое предусловие (например, соединение между элементом системы и атакующим) для возможности реализации $i$- -го атакующего действия из множества $AA$.

Реализация модели $AA$ основана на OWL.

Блок $cps$ может быть представлен следующим образом:

$$bb = (BB^*, HW, SW, P_{bb}) \mid bb \in BB, \quad (4)$$

где $BB^*$ – множество подблоков $bb^*$ блока $bb$; $HW$ – множество аппаратных элементов $bb$ (множество может быть пустым); $SW$ – множество программных элементов $bb$ (множество может быть пустым); $P_{bb}$ – множество свойств $bb$.

Реализация модели $bb$ основана на SYSML-диаграммах.

Аппаратный элемент из множества $HW$ может быть представлен следующим образом:

$$hw = (HW^*, P_{hw}) \mid hw \in HW, \quad (5)$$

где $HW^*$ – множество аппаратных подэлементов $hw^*$ аппаратного элемента $hw$; $P_{hw}$ – множество свойств $hw$.

Реализация модели $hw$ основана на VHDL (VHSIC, *от англ.* Very High Speed Integrated Circuits + Hardware Description Language – язык описания аппаратуры для высокоскоростных интегральных схем) [25].

Программный элемент из множества $SW$ может быть представлен следующим образом:

$$sw = (SW^*, P_{sw}) \mid sw \in SW, \quad (6)$$

где $SW^*$ – множество подэлементов $sw^*$ программного элемента $sw$; $P_{sw}$ – множество свойств $sw$. Реализация модели $sw$ основана на UML (*от англ.* Unified Modeling Language – унифицированный язык моделирования) [26].

Сеть передачи данных между элементами $cps$ может быть представлена следующим образом:

$$nw = (tpl, dte, P_{nw}), \quad (7)$$

где $tpl$ – топология $nw$; $dte$ – среда передачи данных $nw$; $P_{nw}$ – множество свойств $nw$.

Реализация модели $nw$ основана на мультиагентном моделировании с помощью SPARK (*от англ.* Simple Platform for Agent-Based Representation of Knowledge – простая платформа для агентского представления знаний) [27]. Реализация модели $tpl$ основана на языке представления направленных графов O³PRM (*от англ.* Open Object-Oriented Probabilistic Relational Models – открытые объектно-ориентированные вероятностные реляционные модели) [28]. Реализация модели $dte$ основана на представлении графов событий в SIGMA [29].

Взаимосвязь элементов $cps$ обеспечивается за счет их влияния на свойства друг друга. Это означает, что для обеспечения необходимого уровня защищенности КФС целью фазы проектирования системы является поиск наиболее рационального элементного состава $cps$. Наиболее рациональный элементный состав выбирается на основе компро-

мисса между тем, что элементам системы необходимо для их корректной работы (выражается через функциональные требования $FR$ и нефункциональные ограничения $NFL$), и тем, что данные элементы могут системе предоставить (выражается через предоставляемый функционал $PRF$ и предоставляемые ресурсы $PRR$). Кроме того, влияние каждого атакующего действия $aa$ выражается через его влияние на предоставляемый системой $cps$ или ее элементами функционал (отказ в обслуживании) или через его влияние на их потребности (истощение ресурсов). В свою очередь атакующий $a$, в зависимости от своего типа $tp$ и уровня $lvl$, ограничивает множество потенциально реализуемых атакующих действий $AA$ на систему.

Таким образом, свойство из множества $P$ системы $cps$ или ее элемента может быть представлено следующим образом:

$$p = (FR, NFL, PRF, PRR), \quad (8)$$

где $FR$ – множество функциональных требований, удовлетворение которым необходимо для корректной работы $cps$ или ее элементов; $NFL$ – множество нефункциональных ограничений, удовлетворение которым необходимо для корректной работы $cps$ и ее элементов; $PRF$ – множество функциональных возможностей, предоставляемых $cps$ или ее элементами; $PRR$ – множество ресурсов, предоставляемых $cps$ или ее элементами. Общая структура свойства $p$ представлена на рисунке 2.

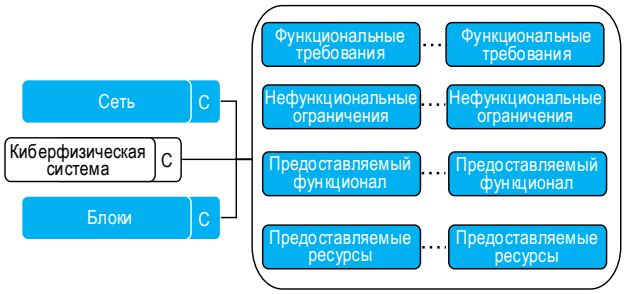


**Рис. 2. Представление свойств в комплексной модели КФС**

Важно отметить, что при расчете свойств системы $cps$ также важно учитывать эмерджентные свойства, возникающие в результате взаимодействия ее элементов. В рамках предлагаемой модели влияние эмерджентных свойств выражается через специальные модификаторы, которые влияют на значения свойств $cps$ и ее элементов во время их взаимодействия (к примеру, взаимодействие различных элементов системы для решения общей задачи нуждается в выделении дополнительных ресурсов на согласование их работы). Процесс влияния эмерджентных свойств может быть представлен следующим образом:

$$f_p(x) = P_x, x = (y_1, \dots, y_n) \mid n \in N,$$
$$f_p(x) = f_p(y_1)EP_x^{y_1} \cup \dots \cup f_p(y_n)EP_x^{y_n}, \quad (9)$$

где $x$ – элемент киберфизической системы $cps$ (или сама $cps$), состоящий из множества подэлементов $y_i \mid i \in 1 \dots n$ (также элементы $cps$); $P_x$ – свойства $x$; $EP_x^{y_i} \mid i \in 1 \dots n$ – эмерджентные свойства $x$.

Влияние эмерджентных свойств $ep_i \in EP \mid i \in 1 \dots n$ может быть классифицировано по области их влияния и результату влияния. По области влияния, эмерджентные свойства $ep_i$ могут быть разделены на аппаратные (влияют на $hw$), программные (влияют на $sw$), программно-аппаратные (влияют на $bb$), топологические (влияют на $tpl$), среды (влияют на $dte$), сетевые (влияют на $nw$) и системные (влияют на $cps$). По результату влияния, эмерджентные свойства $ep_i$ могут быть разделены на положительные (взаимодействие элементов делает решение более рациональным), нейтральные и негативные (взаимодействие элементов делает решение менее рациональным). Отметим, что нейтральные эмерджентные свойства по своей сути отражают их отсутствие, однако, необходимы для полноты классификации.

Процесс верификации в рамках предлагаемой комплексной модели КФС состоит из двух основных частей:

1) проверки реализуемости защищенной $cps$ в соответствии с пожеланиями заказчика, выраженными в виде требований и ограничений, а также информации о доступных ресурсах;

2) проверки защищенности $cps$ относительно атакующего $a$ определенного типа $tp$ и уровня $lvl$ (так-же определяется заказчиком).

## 4. Экспериментальная проверка

Экспериментальная проверка предложенной модели осуществлялась на прототипе защищенной системы контроля и управления доступом (СКУД), который был представлен в работе [30]. Данный прототип состоит из пяти основных частей: устройств на основе микроконтроллеров; сервера приложений; сервера журналирования; приложения администратора и агентов рабочих станций (рисунок. 3). Рассмотрим представление данного прототипа в рамках комплексной модели более подробно.

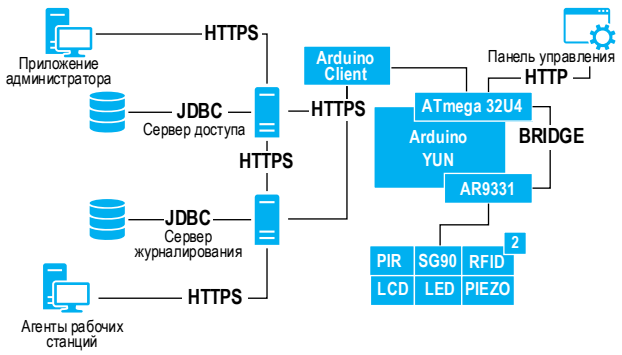


**Рис. 3. Прототип СКУД, представленный в работе [30]**

Разработанный прототип СКУД представляет собой КФС $cps$ с функциональными возможностями организации физического контроля доступа (кому, когда и в какое помещение разрешено входить) и контроля доступа к персональным компьютерам сотрудников.

Физический контроль доступа осуществляется множеством блоков системы $BB_1$, в рамках которого каждый блок $bb_{1i}$ представляет собой устройство на основе микроконтроллера, собранное на основе следующих компонентов:

– микроконтроллер Arduino Yun - подблок блока $bb_{1i}$, состоящий из аппаратных (ATmega 32U4, AR9331 и т. д.) и программных (прошивка, софт, библиотеки) элементов;

– microSD 512 MB, $PRF$ которой расширяет объем доступной микроконтроллеру Arduino Yun флеш-памяти;

– сервопривод TowerPro SG90, $PRF$ которого предоставляет $bb_{1i}$ возможность автоматически открывать и закрывать двери;

– считыватель RFID-карт Grove 125KHz, $PRF$ которого позволяют $bb_{1i}$ проверять уникальные идентификаторы карт пользователей;

– инфракрасный датчик движения HC-SR 501, $PRF$ которого позволяют $bb_{1i}$ обнаруживать попытки несанкционированного доступа;

– другие компоненты, $PRF$ которых позволяет $bb_{1i}$ взаимодействовать с пользователем посредством текстовых сообщений, звуковых и световых сигналов.

Каждый блок $bb_{1i}$ соединен с блоком $bb_2$ (сервер приложений) и блоком $bb_3$ (сервер журналирования) посредством TCP/IP-соединения в сети $nw$. Соединение между множеством блоков $BB_1$ и блоками $bb_2$ и $bb_3$ организовано с помощью Wi-Fi-среды передачи данных $dte$ с WPA2 шифрованием и надежным паролем ($PRF$). Топология $tpl$ сети – звезда (каждый блок $bb_{1i}$ соединен с блоком $bb_2$ и блоком $bb_3$ посредством собственного клиента).

В рамках модели тип $tp$ атакующего $a$ может быть от 0 до 4.

*Тип 0*. Отсутствие у атакующего доступа к множеству блоков $BB$ или сети $nw$ СКУД, а потому влиять на безопасность системы он может только косвенно (атаки с использованием методов социальной инженерии).

*Тип 1*. Наличие у атакующего удаленного доступа к множеству блоков $BB$ или сети $nw$ СКУД (атаки через TCP/IP-соединение с блоками $bb_2$ и $bb_3$).

*Тип 2*. Наличие у атакующего доступа к множеству блоков $BB$ или сети $nw$ СКУД за пределами контролируемого периметра (атаки через Wi-Fi-соединение на множество блоков $BB_1$).

*Тип 3*. Наличие у атакующего физического доступа к множеству блоков $BB$ или сети $nw$ СКУД (атаки через USB-порт на блоки $bb_{1i}$, изменение компонентного состава устройств).

*Тип 4*. Наличие у атакующего полного доступа к множеству блоков $BB$ или сети $nw$ СКУД (атаки через внутренние интерфейсы устройств $bb_{1i}$ (память, система обновления, система отладки), изменение их интегральных схем).

Уровень $lvl$ атакующего $a$ может быть от 1 до 3.

*Уровень 1*. Отсутствие у атакующего специальных знаний о множестве блоков $BB$ или сети $nw$ СКУД. Атакующий $a$ может применять только общеизвестные программные инструменты и эксплуатировать только известные уязвимости (примеры из множества атакующих действий $AA$: атаки на веб-сервер, социальная инженерия, перехват траффика).

*Уровень 2*. Наличие у атакующего $a$ специальных знаний о множестве блоков $BB$ или сети $nw$ СКУД. Атакующий может использовать специализированные инструменты и эксплуатировать уязвимости нулевого дня (примеры из множества атакующих действий $AA$: атаки типа «человек-посередине», отказ в обслуживании, переполнение буфера).

*Уровень 3*. Группа атакующих $A$ 2-го уровня (примеры из множества атакующих действий $AA$: криптографический анализ сообщений, атаки на систему аутентификации, перехват, изменение и подделка сообщений).

Рассмотрим атакующего $a$ 2-го типа 2-го уровня. Множество атакующих действий $AA$, предусловием $prcnd$ которых является наличие Wi-Fi-соединения, выглядит следующим образом:

– перехват;

– изменение и подделка сообщений между устройствами $bb_{1i}$ и серверами $bb_2$, $bb_3$ («человек-посередине»);

– стандартные сетевые атаки на устройства $bb_{1i}$ (отказ в обслуживании, TCP SYN Flood);

– атаки, связанные с эксплуатацией уязвимостей, специфичных для устройств $bb_{1i}$ (отправка некорректных пакетов данных, переполнение буфера);

– криптографический анализ сообщений между устройствами $bb_{1i}$ и серверами $bb_2$, $bb_3$;

– атаки на систему обновления устройств $bb_{1i}$.

Атаки типа «человек-посередине» могут привести к перехвату, изменению и подделке сообщений между устройствами $bb_{1i}$ и серверами $bb_2$ и $bb_3$, если используемое в рамках системы Wi-Fi-соединение используется не только для нужд системы (например, СКУД подключена к Wi-Fi-соединению, которое используют работники организации). В подобной ситуации, если атакующему $a$ удастся получить доступ к сети $nw$ организации, то станет возможным беспрепятственный сбор трафика между устройствами $bb_{1i}$ и серверами $bb_2$, $bb_3$. Это означает потенциальную возможность наводнить трафик системы $cps$ поддельными или измененными сообщениями о событиях от датчиков.

Для предотвращения подобных ситуаций необходимо проанализировать количество незадействованных ресурсов на устройствах $bb_{1i}$. Объем флеш-памяти для хранения прошивки в Arduino Yun составляет 32 КВ, в то время как объем прошивки (проверка доступа, работа с датчиками и передача данных) составляет 17.2 КВ. Это означает, что существует возможность расширить прошивку устройства за счет дополнительных программных элементов, суммарный объем которых не превышает 14.8 КВ. К примеру, прошивку можно расширить за счет программного элемента sw, функциональные возможности $PRF$ которого позволят шифровать передаваемые сообщения. Среда передачи данных $dte$, улучшенная таким образом, позволит обнаружить атаки типа «человек-посередине» за счет аномального роста количества непринятых сообщений от устройств $bb_{1i}$, отправленных на серверы $bb_2$ и $bb_3$ (все сообщения, измененные или подделанные атакующим $a$, будут отброшены серверами, при условии, что алгоритм и ключ шифрования не были раскрыты).

По итогам проведенного эксперимента безопасность соединения между элементами СКУД была улучшена еще на ранних стадиях проектирования. По результатам верификации, проведенным в среде UPPAAL, модель СКУД после внесенных изменений является защищенной относительно злоумышленника $a$ 2-го типа 2-го уровня.

## 5. Область применения

В данной работе представлена новая модель, в рамках которой КФС отображены в виде множества взаимодействующих блоков с определенными свойствами. Данная модель разработана для проектирования защищенных систем и верификации результатов проектирования. Она позволяет отобразить как статическое состояние КФС, так и ее динамику. При этом, в зависимости от отображаемого состояния системы, используются соответствующие подходы и инструменты. Одной из главных сложностей при построении подобной модели является объединение различных подходов к моделированию в рамках единого решения, а также обеспечение возможности преобразования используемых моделей друг в друга без потери значимых данных.

Важно отметить, что предложенное решение ориентировано на обеспечение защищенности КФС. Ресурсы КФС, доступные при предлагаемом подходе к проектированию, представляют собой незадействованные системой ресурсы, а значит, они могут быть использованы для улучшения защищенности КФС. При этом объем ресурсов, используемых КФС в различные моменты времени, не является постоянным. Важно принимать это во внимание при объединении различных блоков системы. К примеру, если два блока системы нуждаются в одинаковом объеме ресурсов для полноценной работы, но никогда не задействуют данные ресурсы одновременно, то и потенциальный конфликт между данными блоками не представляется возможным. С другой стороны, атака спланированная таким образом, чтобы критически важному элементу КФС не хватило ресурсов в нужный момент, может привести к выводу системы из строя.

Сравнение предложенной модели (далее обозначено как «М») с другими комплексными моделями, а именно: Hu F. et al. [2] и Penas O. et al. [12], – в контексте проектирования и верификации защищенных КФС представлено в таблице 1.

Важно отметить, что использование обозначения «+» для решений, рассмотренных в работах [2, 12] и не ориентированных на обеспечение защищенности систем, означает не наличие в данных решениях готовой реализации подобного функционала, а то, что предложенные модели структурно позволяют реализовать на их основе обнаружение соответствующих атакующих действий в процессе проектирования и верификации КФС.

**ТАБЛИЦА 1. Сравнение моделей для проектирования и верификации защищенных КФС**

| Тип | Атакующие действия | [2] | [12] | М |
|---|---|---|---|---|
| 0 | Социальная инженерия | – | – | +/– |
| 1 | Атакующие действия типа 0, «человек посередине», DDoS, TCP SYN Flood, отправка некорректных сетевых пакетов, переполнение буфера, криптографический анализ, атака на систему обновления (для TCP/IP) | + | + | + |
| 2 | Атакующие действия типа 1, но также для IR, Wi-Fi, Bluetooth, атаки по сторонним каналам передачи данных на основе анализа электромагнитного излучения | + | + | + |
| 3 | Атакующие действия типа 2, атаки по сторонним каналам передачи данных, основанные на различных параметрах (прямой доступ), атаки на интерфейсы, замена элементов КФС | + | – | + |
| 4 | Атакующие действия типа 3, дизассемблирование элементов КФС, эксплуатация уязвимостей аппаратных элементов (внутренние интерфейсы, скрытые порты, среда взаимодействия компонентов), изменение данных электронных компонентов, извлечение криптографических ключей | – | – | + |

Кроме того, ни один из проанализированных подходов не рассматривает социальный аспект безопасности КФС в полной мере. В рамках дальнейших исследований планируется расширить предложенную комплексную модель за счет введения пользователей системы с их свойствами, а также определить границы их влияния на работу системы.

## 6. Заключение

В данной работе предложена комплексная модель защищенных КФС, которая представляет собой ключевой элемент представленной ранее методики проектирования и верификации защищенных КФС [31]. Корректность представленной модели обоснована посредством ее применения для проектирования и верификации защищенной системы контроля и управления доступом. В процессе проектирования выявлены атакующие действия, реализация которых потенциально возможна в данной системе с учетом указанных ограничений. В качестве необходимых улучшений для обеспечения защищенности системы предложено расширить функциональные возможности устройств на основе микроконтроллеров за счет добавления в прошивку алгоритма шифрования передаваемых данных.

В рамках дальнейших исследований планируется проанализировать различные каталоги для расширения базы знаний по атакующим действиям на КФС, а также классифицировать их. Кроме того, планируется проведения экспериментов в других областях приложения (например, железнодорожная инфраструктура и робототехнические комплексы).

**БЛАГОДАРНОСТИ**

*Исследование выполнено при частичной финансовой поддержке РФФИ в рамках научного проекта № 19-37-90082 и бюджетной темы 0073-2019-0002.*

**Список используемых источников**

1. Левшун Д.С., Чечулин А.А., Котенко И.В. Проектирование безопасной среды передачи данных на примере протокола I2C // Защита информации. Инсайд. 2018. № 4 (82). С.54–62.

2. 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

3. Canedo A., Schwarzenbach E., Faruque M.A.A. Context-sensitive synthesis of executable functional models of cyber-physical systems // Proceeding of the International Conference on Cyber-Physical Systems (ICCPS, Philadelphia, USA, 8–11 April 2013). Piscataway, NJ: IEEE, 2013. PP. 99–108. DOI:10.1145/2502524.2502539

4. Srivastava A., Morris T., Ernster T., Vellaithurai C., Pan S., Adhikari U. Modeling Cyber-Physical Vulnerability of the Smart Grid With Incomplete Information // IEEE Transactions on Smart Grid. 2013. Vol. 4. Iss. 1. PP. 235–244. DOI:10.1109/TSG.2012.2232318

5. Xinyu C., Huiqun Y., Xin X. Verification of Hybrid Chi Model for Cyber-Physical Systems Using PHAVer // Proceeding of Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (Taichung, Taiwan, 3–5 July 2013). Piscataway, NJ: IEEE, 2013. PP. 122–128. DOI:10.1109/IMIS.2013.29

6. Nuzzo P., Sangiovanni-Vincentelli A.L., Bresolin D., Geretti L., Villa T. A Platform-Based Design Methodology With Contracts and Related Tools for the Design of Cyber-Physical Systems // Proceedings of the IEEE. 2015. Vol. 103. Iss. 11. PP. 2104–2132. DOI:10.1109/JPROC.2015.2453253

7. Selic B. The pragmatics of model-driven development // IEEE Software. 2003. Vol 20. Iss. 5. PP. 19–25. DOI:10.1109/MS.2003.1231146

8. Sztipanovits J., Karsai G. Model-integrated computing // Computer. 1997. Vol. 30. Iss. 4. PP. 110–111. DOI:10.1109/ 2.585163

9. Kelly S., Tolvanen J.-P. Domain-Specific Modeling: Enabling Full Code Generation. Hoboken: John Wiley & Sons, 2008.

10. Hehenberger P., Vogel-Heuser B., Bradley D., Eynard B., Tomiyama T., Achiche S. Design, modelling, simulation and integration of cyber physical systems: Methods and applications // Computers in Industry. 2016. Vol. 82. PP. 273–289. DOI:10.1016/j.compind.2016.05.006

11. Fritzson P. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Hoboken: John Wiley & Sons, 2010.

12. 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

13. Friedenthal S., Alan M. and Rick S. A Practical Guide to SysML: The Systems Modeling Language. Burlington: Morgan Kaufmann, 2014.

14. Brück D., Elmqvist H., Olsson H., Mattsson S.E. Dymola for Multi-Engineering Modeling and Simulation. Proceedings of the 2nd International Modelica Conference (Oberphaffenhofen, Germany, 18–19 March 2002). 2002.

15. Estefan J.A. Survey of Model-Based Systems Engineering (MBSE) Methodologies. URL: http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf (Accessed 21 November 2019)

16. Rumbaugh J., Blaha M., Premerlani M., Eddy F., Lorensen W. Object-Oriented Modeling and Design. Englewood Cliffs: Prentice-Hall, 1991.

17. McGuinness D.L., van Harmelen F. OWL Web Ontology Language Overview // W3C recommendation. 2004.

18. Balasubramaniyan S., Srinivasan S., Buonopane F., Subathra B., Vain J., Ramaswamy S. Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL // Microprocessors and microsystems. 2016. Vol. 42. PP. 37–48. DOI:10.1016/j.micpro.2015.12.006

19. Ohlin M., Henriksson D., Cervin A. TrueTime 1.5 – Reference Manual. Department of Automatic Control, Lund Institute of Technology, Lund University, 2007.

20. Larsen K.G., Pettersson P., Yi W. UPPAAL in a nutshell // International Journal on Software Tools for Technology Transfer (STTT). 1997. Vol. 1. Iss. 1. PP. 134–152.

21. Seiger R., Keller C., Niebling F., Schlegel T. Modelling complex and flexible processes for smart cyber-physical environments // Journal of Computational Science. 2015. Vol. 10. PP. 137–148. DOI:10.1016/j.jocs.2014.07.001

22. Steinberg D., Budinsky F., Merks E., Paternostro M. EMF: Eclipse Modeling Framework. London: Pearson Education, 2008.
23. Srinivasan S., Buonopane F., Vain J., Ramaswamy S. Model checking response times in Networked Automation Systems using jitter bounds // Computers in Industry. 2015. Vol. 74. PP. 186–200. DOI:10.1016/j.compind.2015.06.012
24. Goldblatt R. Logics of Time and Computation. Stanford: Center for the Study of Language and Information, 1992.
25. Zainalabedin N. VHDL: Analysis and Modeling of Digital Systems. New York: McGraw-Hill, 1997.
26. Fowler M., Scott K. UML Distilled: a Brief Guide to the Standard Object Modeling Language. Boston: Addison-Wesley Professional, 2004.
27. Solovyev A., Mikheev M., Zhou L., Dutta-Moscato J., Ziraldo C., An G., et al. SPARK: a framework for multi-scale agent-based biomedical modeling // Proceedings of the Spring Simulation Multiconference (Orlando, USA, 11–15 April 2010). San Diego: Society for Computer Simulation International, 2010. DOI:10.1145/1878537.1878541
28. Torti L., Wuillemin P. O3PRM Language Specification. Technical report UPMC. 2013.
29. Schruben L.W. SIGMA – A graphical approach to teaching simulation // Journal of Computing in Higher Education. 1992. Vol. 4. DOI:10.1007/BF02940978
30. Десницкий В.А., Чечулин А.А., Котенко И.В., Левшун Д.С., Коломеец М.В. Комбинированная методика проектирования защищенных встроенных устройств на примере системы охраны периметра // Труды СПИИРАН. 2016. № 5(48). С. 5–31. DOI: 10.15622/sp.48.1
31. Левшун Д.С., Чечулин А.А., Котенко И.В. Жизненный цикл разработки защищенных систем на основе встроенных устройств // Защита информации. Инсайд. 2017. № 4(76). С. 53–59.

* * *

# A COMBINED MODEL OF SECURE CYBER-PHYSICAL SYSTEMS FOR THEIR DESIGN AND VERIFICATION

**D. Levshun**[1, 2], **A. Chechulin**[1, 3], **I. Kotenko**[1, 3]

[1]St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science,
St. Petersburg, 199178, Russia
[2]St. Petersburg National Research University of Information Technologies, Mechanics and Optics,
St. Petersburg, 197101, Russia
[3]The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

**Article info**
The article was received 05th November 2019

**For citation:** Levshun D., Chechulin A., Kotenko I. A Comprehensive Model of Secure Cyber-Physical Systems for their Design and Verification. *Proceedings of Telecommunication Universities.* 2019;5(4):114–123. (in Russ.) Available from: https://doi.org/10.31854/1813-324X-2019-5-4-114-123

**Abstract:** *In this paper a combined model of secure cyber-physical systems for their design and verification is proposed. Within the framework of this model, a cyber-physical system is represented as a set of blocks with various properties and relationships between them. The main challenge in such model construction is to combine various approaches to the modeling of cyber-physical systems (or their elements) within a single approach. The main goal of the proposed modeling approach is to provide the ability to convert various models into each other without losing significant data about the elements of the system, as well as taking into account the emergent properties that arise in the process of their interaction. The correctness of the proposed model is validated by the example of its use for design and verification of access control system.*

**Keywords:** *security by design, cyber-physical system, security verification, system modeling, attacker model, attack actions model, comprehensive model.*

**References**

1. Levshun D.S., Chechulin A.A., Kotenko I.V. Design of Secure Data Transfer Environment Using the I2C Protocol as an Example. *Information Security. Inside.* 2018;4(82):54–62. (in Russ.)
2. 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. Available from: https://doi.org/10.1016/j.future.2015.06.006

3. Canedo A., Schwarzenbach E., Faruque M.A.A. Context-sensitive synthesis of executable functional models of cyber-physical systems. *Proceedings of the International Conference on Cyber-Physical Systems, ICCPS, 8–11 April 2013, Philadelphia, USA*. Piscataway, NJ: IEEE; 2013. p.99–108. Available from: https://doi.org/10.1145/2502524.2502539

4. Srivastava A., Morris T., Ernster T., Vellaithurai C., Pan S., Adhikari U. Modeling Cyber-Physical Vulnerability of the Smart Grid With Incomplete Information. *IEEE Transactions on Smart Grid*. 2013;4(1):235–244. Available from: https://doi.org/10.1109/TSG.2012.2232318

5. Xinyu C., Huiqun Y., Xin X. Verification of Hybrid Chi Model for Cyber-Physical Systems Using PHAVer. *Proceeding of Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 3–5 July 2013, Taichung, Taiwan*. Piscataway, NJ: IEEE; 2013. p.122–128. Available from: https://doi.org/10.1109/IMIS.2013.29.

6. Nuzzo P., Sangiovanni-Vincentelli A.L., Bresolin D., Geretti L., Villa T. A Platform-Based Design Methodology With Contracts and Related Tools for the Design of Cyber-Physical Systems. *Proceedings of the IEEE*. 2015;103(11):2104–2132. Available from: https://doi.org/10.1109/JPROC.2015.2453253

7. Selic B. The pragmatics of model-driven development. *IEEE Software*. 2003;20(5):19–25. Available from: https://doi.org/ 10.1109/ MS.2003.1231146

8. Sztipanovits J., Karsai G. Model-integrated computing. *Computer*. 1997;30(4):110–111. Available from: https://doi.org/10.1109/2.585163

9. Kelly S., Tolvanen J.-P. *Domain-Specific Modeling: Enabling Full Code Generation*. Hoboken: John Wiley & Sons; 2008.

10. Hehenberger P., Vogel-Heuser B., Bradley D., Eynard B., Tomiyama T., Achiche S. Design, modelling, simulation and integration of cyber physical systems: Methods and applications. *Computers in Industry.* 2016;82:273–289. Available from: https://doi.org/10.1016/j.compind.2016.05.006

11. Fritzson P. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1.* Hoboken: John Wiley & Sons; 2010.

12. 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. Available from: https://doi.org/10.1016/j.compind.2016.12.001

13. Friedenthal S., Alan M., Rick S. A Practical Guide to SysML: The Systems Modeling Language. Burlington: Morgan Kaufmann, 2014.

14. Brück D., Elmqvist H., Olsson H., Mattsson S.E. Dymola for Multi-Engineering Modeling and Simulation. Proceedings of the 2nd International Modelica Conference, 18–19 March 2002, Oberphaffenhofen, Germany. 2002.

15. Estefan J.A. *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. Available from: http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf [Accessed 21 November 2019]

16. Rumbaugh J., Blaha M., Premerlani M., Eddy F., Lorensen W. *Object-Oriented Modeling and Design*. Englewood Cliffs: Prentice-Hall; 1991.

17. McGuinness D.L., van Harmelen F. OWL Web Ontology Language Overview. *W3C recommendation.* 2004.

18. Balasubramaniyan S., Srinivasan S., Buonopane F., Subathra B., Vain J., Ramaswamy S. Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL. *Microprocessors and microsystems*. 2016;(42):37–48. Available from: https://doi.org/10.1016/j.micpro.2015.12.006

19. Ohlin M., Henriksson D., Cervin A. *TrueTime 1.5 – Reference Manual*. Department of Automatic Control, Lund Institute of Technology, Lund University, 2007.

20. Larsen K.G., Pettersson P., Yi W. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*. 1997;1(1):134–152.

21. Seiger R., Keller C., Niebling F., Schlegel T. Modelling complex and flexible processes for smart cyber-physical environments. *Journal of Computational Science*. 2015;10:137–148. Available from: https://doi.org/10.1016/j.jocs.2014.07.001

22. Steinberg D., Budinsky F., Merks E., Paternostro M. *EMF: Eclipse Modeling Framework*. London: Pearson Education, 2008.

23. Srinivasan S., Buonopane F., Vain J., Ramaswamy S. Model checking response times in Networked Automation Systems using jitter bounds. *Computers in Industry*. 2015;74:186–200. Available from: https://doi.org/10.1016/j.compind.2015.06.012

24. Goldblatt R. Logics of Time and Computation. Stanford: Center for the Study of Language and Information; 1992.

25. Zainalabedin N. *VHDL: Analysis and Modeling of Digital Systems*. New York: McGraw-Hill; 1997.

26. Fowler M., Scott K. *UML Distilled: a Brief Guide to the Standard Object Modeling Language*. Boston: Addison-Wesley Professional; 2004.

27. Solovyev A., Mikheev M., Zhou L., Dutta-Moscato J., Ziraldo C., An G., et al. SPARK: a framework for multi-scale agent-based biomedical modeling. *Proceedings of the Spring Simulation Multiconference, 11–15 April 2010, Orlando, USA*. San Diego: Society for Computer Simulation International; 2010. Available from: https://doi.org/10.1145/1878537.1878541

28. Torti L., Wuillemin P. *O3PRM Language Specification*. Technical report UPMC. 2013.

29. Schruben L.W. SIGMA – A graphical approach to teaching simulation. *Journal of Computing in Higher Education*. 1992;4. Available from: https://doi.org/10.1007/BF02940978

30. Desnitsky V.A., Chechulin A.A., Kotenko I.V., Levshun D.S., Kolomeec M.V. Combined Design Technique for Secure Embedded Devices Exemplified by a Perimeter Protection System. *SPIIRAS Proceedings*. 2016;5(48):5–31. (in Russ.) Available from: https://doi.org/10.15622/sp.48.1

31. Levshun D.S., Chechulin A.A. and Kotenko I.V. Design Lifecycle for Secure Cyber-Physical Systems Based on Embedded Devices. *Information Security. Inside.* 2017;4(76):53–59. (in Russ.)