УДК 004.056.53 **DOI** 10.15622/sp.48.1

В.А. Десницкий, А.А. Чечулин, И.В. Котенко, Д.С. Левшун, М.В. Коломеец

КОМБИНИРОВАННАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ ЗАЩИЩЕННЫХ ВСТРОЕННЫХ УСТРОЙСТВ НА ПРИМЕРЕ СИСТЕМЫ ОХРАНЫ ПЕРИМЕТРА

Десницкий В.А., Чечулин А.А., Котенко И.В., Левшун Д.М., Коломеец М.В. **Комбинированная методика проектирования защищенных встроенных устройств на примере системы охраны периметра.**

Аннотация. С точки зрения информационной безопасности встроенные устройства представляют собой элементы сложных киберфизических систем, работающих в потенциально враждебном окружении. Поэтому разработка таких устройств является сложной задачей, часто требующей экспертных решений. Сложность задачи разработки защищенных встроенных устройств обуславливается различными типами угроз и атак, которым может быть подвержено устройство, а также тем, что на практике вопросы безопасности встроенных устройств обычно рассматриваются на финальной стадии процесса разработки в виде добавления дополнительных функций защиты. В статье предлагается методика проектирования, применение которой будет способствовать разработке безопасных и энергоэффективных киберфизических и встроенных устройств. Данная методика организует поиск наилучших комбинаций компонентов защиты на основе решения оптимизационной задачи. Работоспособность предлагаемой методики демонстрируется на основе разработки прототипа защищенной системы охраны периметра помещения.

Ключевые слова: встроенные устройства, киберфизические системы, охрана периметра, проектирование защищенных киберфизических систем.

1. Введение. В настоящее время встроенные устройства получают все большее распространение в самых разных областях приложения — в системах управления и контроля на транспорте, в системах управления производственным процессом, системах, предоставляющих телекоммуникационные сервисы потребителям, в системах имплантируемых медицинских устройств для контроля жизненно важных показаний организма человека, в электроэнергетике, в системах обеспечения физической безопасности помещений, прикладных системах распознавания речи и др. Критически важный характер таких систем, а также высокая степень взаимодействия встроенного устройства с другими элементами программно-аппаратного окружения и пользователями системы обуславливает важность разработки механизмов защиты таких устройств от угроз информационной безопасности.

Под встроенным устройством понимается электронное устройство, функциональность которого определяется прежде всего его аппаратной и программной частями. Аппаратная составляющая определяет его вычислительные и коммуникационные возможности, интерфейсы взаимодействия с источниками данных (сенсорами) и различными сило-

выми приводами, производительность, энергоэффективность, автономность, мобильность и другие возможные характеристики устройств. Программная же часть реализует бизнес-логику устройства с использованием драйверов и библиотек для связи с периферийными аппаратными модулями, базами данных, веб-сервисами и др. Ключевым признаком встроенного устройства является его узкоспециализированное назначение, причем, как правило, такие устройства имеют следующие особенности: 1) жесткие ограничения на аппаратные возможности устройств и энергоресурсы, обуславливаемые, в частности, использованием одноплатных компьютеров; 2) изменчивость киберфизического окружения встроенных устройств, и как следствие, их подверженность специфичным наборам атак; 3) компонентно-ориентированная структура с возможными скрытыми конфликтами между отдельными компонентами или их побочным влиянием друг на друга.

Приведенные выше особенности встроенных устройств обуславливают потребность в разработке специализированных подходов и методов проектирования, которые позволили бы повысить защищенность конечных продуктов и сервисов [1].

Данная работа сфокусирована на решении вопросов проектирования безопасных систем в части комбинирования отдельных компонентов встроенных устройств в единый, согласованно работающий программно-аппаратный комплекс. Такое комбинирование осуществляется на основе установленных требований к защите с учетом ограничений устройств и связей между компонентами. Данная статья является логическим продолжением работ, опубликованных ранее по проектированию и верификации систем со встроенными устройствами. В [2, 3] были предложены общие рекомендации по комбинированию компонентов защиты систем со встроенными устройствами с учетом показателей их ресурсопотребления (конфигурированию) с привлечением оптимизационного подхода для комбинаторного перебора имеющихся альтернатив компонентов защиты и эвристического подхода для определения порядка учета показателей ресурсопотребления.

В рамках данной статьи предложена методика проектирования защищенных встроенных устройств на основе комбинирования компонентов защиты в виде структурированной последовательности действий, которые разработчик должен выполнить для формирования эффективной реализации защищенного устройства. Помимо этого, вкладом данной статьи является также подтверждение выполнимости предложенной методики путем ее практического применения при проектировании защищенной системы охраны периметра помещения (в части реализации функций контроля доступа) с использованием одно-

платных компьютеров. При этом практический результат применения методики — набор выбранных программных и программно-аппаратных компонентов из списков имеющихся альтернатив, применение которых позволило построить защищенную систему с учетом улучшения ее целевых показателей, в том числе цены и некоторых показателей ресурсопотребления.

Новизна данной статьи заключается в (1) разработке методики проектирования защищенных встроенных устройств путем комбинирования компонентов защиты с использованием правил их выбора с учетом семантики защитного функционала и имеющихся нефункциональных ограничений, а также (2) подтверждение корректности этой методики путем ее практического применения для создания системы контроля доступа в помещение с использованием программируемых микроконтроллеров.

Данная статья имеет следующую структуру. В разделе 2 приведен обзор работ в предметной области. В разделе 3 представлено описание предлагаемой методики проектирования защищенных встроенных устройств на основе комбинирования компонентов защиты. Разделы 4-6 сфокусированы на проверке корректности предложенной методики: в разделе 4 сформулированы требования к разрабатываемой системе охраны периметра; раздел 5 содержит описание процесса выбора программно-аппаратных средств, при помощи которых может быть реализована система, соответствующая требованиям, определенным в разделе 4; в разделе 6 приведено описание разработанной системы. Раздел 7 включает анализ полученных результатов. Завершает статью заключение, содержащее основные выводы.

2. Работы в предметной области. В соответствии с [4], встроенные устройства определяются как программно-аппаратные устройства, вычислительный процесс которых тесно связан с реакцией на процессы физического окружения и выполняется в рамках некоторой физической платформы, которая, помимо непосредственно вычислительных модулей, включает также модули, взаимодействующие с киберфизическими объектами окружения. К таким объектам относятся разнообразные сенсоры, силовые приводы, сканеры текстовых, звуковых и других данных, устройства отображения информации, разнообразные коммуникационные и навигационные устройства, бытовые и промышленные устройства нагревания, вентиляции, насосные станции, устройства мониторинга и диагностики и др.

Как следствие, связи между программной частью устройства, с одной стороны, и аппаратно-техническим окружением — с другой,

обуславливают наличие дополнительных ограничений, влияющих существенным образом на процесс проектирования таких устройств.

В настоящее время широкое применение на практике получил компонентный подход к проектированию встроенных устройств [5], реализованный в рамках платформ Arduino, Raspberry Pi, Beagle board и операционной системы Android. Фактически, в силу специфики встроенных устройств, завязанных в своей работе непосредственно на обеспечение установленных требований защиты, встроенное устройство представляется в виде множества взаимодействующих программных и программно-аппаратных компонентов. Далее в статье компоненты встроенных устройств, выбор которых во многом определяет выполнимость требований защиты, будем обобщенно называть — компоненты защиты.

На практике при разработке встроенных устройств зачастую выбор тех или иных компонентов защиты осуществляется экспертно, в силу заранее предопределенных субъективных предпочтений разработчиков и уже известных им решений. При этом в процессе комбинирования таких компонентов в единый механизм недостаточное внимание уделяется индивидуальным особенностям компонентов защиты, возможным неявным связям и скрытым конфликтам между компонентами в силу отсутствия их априорной согласованности между собой и ограничениям программно-аппаратной платформы.

Для встроенных устройств, которым присущи как вычисления, так и физические ограничения в [6] также обосновывается важность достижения компромиссов между защищенностью и нефункциональными характеристиками встроенных устройств, в том числе с применением оптимизационных подходов и комбинирования встроенных устройств из отдельных компонент в соответствии с их свойствами и требованиями. При этом учитываются вопросы их корректного взаимодействия в виде последовательного и параллельного функционирования компонентов, что характерно для программных и аппаратных систем, образующих конкретное встроенное устройство.

В [7] обосновывается необходимость и важность исследования вопросов разработки защищенных встроенных устройств на основе использования высокоуровневых средств защиты с приемлемыми энергетическими и вычислительными расходами. Помимо вопросов предоставления устройству и его сервисам необходимых аппаратных и энергоресурсов, особый интерес представляют атаки, направленные на истощение энергоресурсов устройства [8]. При этом подобные атакующие воздействия не обнаружимы посредством традиционно применяемых решений, но обуславливают неконтролируемый расход ресур-

сов со стороны наиболее энергозатратных аппаратных модулей устройства, таких как интерфейсные модули Wi-Fi, Bluetooth и дисплей, и тем самым делая невозможным на некоторое время дальнейшее функционирование устройства. Поэтому комплексная система защиты встроенного устройства должна включать программные и программно-аппаратные модули, направленные против релевантных угроз безопасности с учетом возможных неявных связей между модулями, несогласованностей между ними и побочными эффектами.

В качестве пути достижения компромисса между защищенностью устройства и его ресурсопотреблением в [9] предлагается использование «реконфигурируемых примитивов безопасности» на основе динамической адаптации архитектуры устройства в зависимости от состояния устройства и его окружения. Предлагаемая адаптация основывается, во-первых, на возможности динамического переключения между несколькими механизмами, встроенными в устройство, и во-вторых, на возможности обновления элементов этих механизмов.

В специальной литературе существует достаточно много примеров решения задачи построения системы охраны периметра на основе использования встроенных устройств, объединяемых в единую сеть для обеспечения возможности централизованного управления. Так, например, в работе [10] представлена архитектура, анализ и результаты тестирования распределенной системы контроля доступа. В работе [11] представлена информация о возможных потребителях и отличительных особенностях распределенных систем контроля периметра. Однако в этих работах архитектура разработанных решений основана на экспертном методе.

Методика, представленная в настоящей работе, позволяет выбирать основные решения для построения подобной системы без обязательного вовлечения эксперта в области компонентов защиты встроенных устройств с возможностью автоматизации ее отдельных стадий в части перебора и сравнения большого количества функциональных требований защиты и альтернатив компонентов защиты.

3. Методика проектирования защищенных встроенных устройств. В данном разделе представлена предлагаемая методика проектирования защищенных встроенных устройств на основе комбинирования компонентов защиты. Отличительной особенностью методики является учет функциональных и нефункциональных характеристик компонентов защиты, ограничений устройства и связей между компонентами с использованием оптимизационного подхода.

Под конфигурацией защиты понимается набор компонентов защиты с определенными функциональными и нефункциональными ха-

рактеристиками. Цель методики — определить наиболее эффективную с точки зрения заданных нефункциональных показателей (оптимальную) конфигурацию защиты на основе входных данных об особенностях устройства и возможных компонентах защиты (таблица 1).

Таблица 1. Представление методики проектирования защищенных встроенных устройств

№	Стадии методики
1	Определение функциональных требований к защите
2	Определение нефункциональных ограничений, существенных для проектируемого данного устройства
3	Выявление множества альтернатив компонентов защиты, которые его реализуют, для каждого функционального требования защиты
4	Определение правил выбора компонентов защиты, исходя из связей между ними
5	Вычисление значений нефункциональных показателей для заданных компонентов
6	Упорядочивание альтернатив компонентов по степени ухудшения значений установленных нефункциональных ограничений
7	Определение порядка учета рассматриваемых нефункциональных показателей
8	Исследование альтернатив компонентов защиты и вычисление суммарных значений нефункциональных показателей наборов компонентов защиты, а также выбор оптимальной конфигурации

Фактически, в рамках методики решается задача дискретной оптимизации на множестве конфигураций с целевой функцией, выражаемой при помощи нефункциональных показателей и ограничений на заданные как функциональные, так и нефункциональные показатели [3]. В качестве целевой функции решаемой оптимизационной задачи рассматривается упорядоченный набор из нескольких нефункциональных показателей, каждый из которых подвергается либо минимизации, либо максимизации, в зависимости от семантики нефункциональной характеристики, лежащей в основе рассматриваемого нефункционального показателя *(p1→min/max, p2→min/max, p3→min/max,…)*. Порядок показателей определяется на основе эвристического подхода.

Первая стадия включает определение функциональных требований защиты, которые нужно реализовать в процессе разработки комбинированного механизма защиты. Данные требования основываются на анализе спецификации целевого устройства с использованием методов аналитического моделирования действий нарушителя [12–14]. В качестве примера можно привести следующее функциональное требование защиты: «секретность бизнес-данных устройства должна осу-

ществляться с использованием симметричного шифрования с ключами не менее 128 бит».

Стадия 2 включает действия по определению нефункциональных ограничений, существенных для проектируемого данного устройства. Источником возможных нефункциональных ограничений является методология MARTE [5], где релевантные нефункциональные показатели, характерные для встроенных устройств, специфицированы с использованием UML. В частности, в рамках методики используются нефункциональные ограничения, построенные на основе следующих классов доменов знаний: HW_Physical, HW_PowerSupply, HW_StorageManager, HW_Computing, HW_Communication [5].

На стадии 3 для каждого функционального требования защиты осуществляется определение множества альтернатив компонентов защиты, которые его реализуют. Например, для требования секретности бизнес-данных определяется набор криптографических алгоритмов симметричного блочного шифрования заданной стойкостью с установленной длиной ключа, таких как AES/128/192/256, IDEA и др.

На стадии 4 осуществляется определение правил выбора компонентов защиты, исходя из связей между ними и учитывая семантику компонентов защиты, установленных требований защиты и сценариев использования. Каждое такое правило представляется в виде формальной четверки, имеющей следующие элементы *(req, Alts, reason, justif)*, где *req* — формулировка функционального требования защиты; *Alts* — набор альтернатив компонентов, каждый из которых реализует данное требование; *reason* — причинно-следственная связь в определении предпочтительности компонентов из *Alts,* в зависимости от рассматриваемых для данного требования нефункциональных показателей (т.е. формулировка критерия выбора); и *justif* — фактическое обоснование предлагаемого порядка предпочтительности компонентов из *Alts* для данного функционального требования защиты.

На стадии 5 производится определение значений нефункциональных ограничений для заданных компонентов защиты следующими способами: путем сбора данных от конкретных производителей используемых программно-аппаратных модулей; эмпирически — на основе программного моделирования компонентов защиты (когда это возможно); экспертно — с учетом предыдущего опыта работы с такими или сходными компонентами. Так, например, для каждого из имеющихся альтернативных алгоритмов удаленной аттестации критических бизнес-данных встроенного устройства определяется величина необходимой оперативной памяти (КБ), которое устройство должно предоставить, и объем коммуникационного ресурса, расходуемого на

передачу аттестующих подписей доверенному серверу в единицу времени (Мбит/сек).

Стадия 6 включает упорядочивание альтернатив компонентов защиты по степени ухудшения значений их нефункциональных ограничений. Фактически, для каждого нефункционального показателя осуществляется сортировка компонентов защиты. Например, для учета энергопотребления имеющихся разновидностей некоторого программно-аппаратного компонента защиты возможные альтернативы упорядочиваются в соответствии с уменьшением величины потребляемого ими тока (измеряемого в миллиамперах).

На стадии 7 определяется порядок учета рассматриваемых нефункциональных ограничений в зависимости от относительной важности каждого из них с использованием эвристики, предложенной в [3]. Данная эвристика задает общий алгоритм приоритезации нефункциональных ограничений встроенного устройства. По существу, для каждого нефункционального ограничения выделяется набор специфичных функциональных и нефункциональных признаков встроенного устройства, таких как «наличие постоянного источника питания», «возможность замены устройства или аккумулятора без ущерба для предоставляемых им сервисов», «степень зависимости достижения бизнес-целей устройства от энергоресурсов» и др. Для каждого такого признака предопределено значение ранга (например, с заданием значений от 1 до 3, где 1 — низкая важность, 3 — высокая важность) в зависимости от критичности данного признака для выполнимости заданного нефункционального ограничения (например, ограничения на ресурс энергопотребления). В результате спецификация целевого встроенного устройства анализируется на предмет наличия у него обозначенных признаков. Для каждого нефункционального ограничения выбирается максимальное значение ранга по всем выявленным у разрабатываемого устройства признакам, в соответствии с которыми происходит упорядочивание уже собственно нефункциональных ограничений. При этом ограничения, получившие одинаковые результирующие значения ранга, упорядочиваются между собой согласно порядку, предопределенному экспертно [3].

На стадии 8 осуществляются комбинаторный перебор альтернатив компонентов защиты и вычисление суммарных значений нефункциональных показателей наборов компонентов защиты (конфигураций). Стадия включает также выбор оптимальной конфигурации на основе полученных значений. В частности, в случае большого числа рассматриваемых функциональных требований защиты и имеющихся альтернатив компонентов защиты на данной стадии целесообразно

применять разработанное программное средство Конфигуратор, позволяющее автоматизировать процесс перебора и вычисления.

В случае если в рамках установленных ограничений решений оптимизационной задачи не существует, на стадии 8 предлагается ряд конфигураций, которые смогут быть реализованными при ослаблении определенных ограничений (в частности, увеличения объемов аппаратных ресурсов устройства, выделяемых на работу компонент).

Отметим, что в общем случае выбор компонентов — итеративный процесс. При каком-либо изменении спецификации системы или особенностей ее реализации итоговый набор компонентов может изменяться вследствие изменений условий, которые учитывались в процессе выбора компонентов.

4. Требования к разрабатываемому прототипу системы охраны периметра. Для демонстрации работоспособности предложенной методики проектирования и разработанных программных средств была выбрана задача построения прототипа системы охраны периметра. Данная система должна осуществлять управление физическим доступом в определенном здании (кого пускать, в какое время пускать и в какое помещение пускать), включая ограничение доступа в заданное помещение и идентификацию лица, имеющего доступ в заданное помещение. Кроме того, система должна контролировать информационный доступ (разрешать или запрещать доступ к информации, расположенной на персональных компьютерах). Каждый пользователь, контролируемый системой, может находиться в четырех состояниях: S_1 — пользователь вошел в помещение; S_2 — пользователь авторизовался на рабочем месте (начало сеанса работы с операционной системой); S_3 — пользователь завершил сеанс работы с операционной системой; S_4 — пользователь покинул помещение.

Вход пользователя в помещение или выход пользователя из помещения идентифицируется приложением бесконтактной карты к считывателю, который подсоединен к подсистеме контроля доступа в помещении. В ситуациях, когда пользователь не приложил карту, открытие двери идентифицируется переходом кнопки из нажатого состояния в свободное и/или инфракрасным датчиком движения. Как только установлено, что человек вошел в помещение (или вышел из него), запускается обратный отсчет — время, за которое пользователю необходимо авторизоваться в системе при помощи карты. На основе уникальных данных карты формируется специальный запрос к центральному серверу управления доступом для получения информации о наличии или отсутствии доступа к помещению у пользователя карты. Подсистема контроля доступа в помещение, получив ответ от цен-

трального сервера управления доступом, начинает обработку полученной информации. Результат проверки карты пользователя сопровождается открытием замка, выводом текстовой информации на экран, световым и звуковым сигналом. Результаты проверки карт вместе с информацией о состоянии сеансов работы пользователей с операционной системой направляются на сервер журналирования.

Таким образом, для реализации защищенной системы охраны периметра в целом необходимо реализовать надежную и защищенную подсистему контроля доступа в помещение. Данная подсистема представляет собой встроенное устройство (ВУ), расположенное в дверях между помещениями и подключенное к механизму управления замком.

Рассмотрим основные функциональные требования к данному устройству более подробно.

ВУ должно осуществлять работу в локальной сети системы по беспроводному каналу передачи данных. Это необходимо для защиты от физического воздействия на канал передачи данных между микроконтроллером и центральным сервером управления (например, повреждение Ethernet-кабеля), а также для существенного снижения стоимости установки системы.

ВУ должно поддерживать интерфейс для взаимодействия с удаленным сервером приложений для удобства интеграции в общую систему контроля и управления доступом. Взаимодействие с сервером приложений может осуществляется по HTTP, HTTPS, SOAP. Данные, передаваемые по HTTP и SOAP, должны быть предварительно зашифрованы.

Для взаимодействия с центральным сервером управления доступом ВУ должно поддерживать запуск приложений, разработанных на одном из высокоуровневых языков программирования.

Разрабатываемое ВУ должно поддерживать аварийный режим работы, в случае, если обмен данными между ВУ и центральным сервером управления доступом перестает быть возможным (отсутствие соединения с сервером, отказ в обслуживании на стороне сервера и т.п.). В аварийном режиме работы решение о предоставлении пользователю доступа в помещение принимается на основе локальной базы данных, расположенной на ВУ. Локальная база данных представляет собой резервную копию сетевой базы данных доступа и содержит информацию об администраторах системы. Таким образом, при аварийном режиме работы, доступ в помещение может получить только сотрудник с правами администратора. При этом разрабатываемое ВУ должно обладать объемом памяти, достаточным для хранения и поддержки локальной базы данных.

ВУ должно предоставлять веб-интерфейс для управления ВУ при локальном Ethernet подключении. Данный веб-интерфейс должен содержать внутренний журнал ВУ и обеспечить инициализацию ре-

зервного копирования сетевой базы данных доступа. Таким образом, функциональные требования могут быть сведены в общее представление (таблица 2).

Таблица 2. Функциональные требования к разрабатываемому ВУ

Функциональные требования	№	Описание
К аппаратному обеспечению	1	Обеспечение взаимодействия с внешними электронными компонентами: механическими замками, сканерами бесконтактных RFID-карт, инфракрасными датчиками движения, устройствами вывода текстовой и звуковой информации, звуковых и световых сигналов.
	2	Обеспечение беспроводного канала передачи данных.
	3	Обеспечение передачи данных через Ethernet.
К программному обеспечению	4	Обеспечение обмена данными по HTTP, HTTPS, SOAP.
	5	Обеспечение запуска приложений, написанных на JAVA, Python, C++.
	6	Обеспечение хранения локальной резервной копии базы данных.
	7	Обеспечение шифрования данных, передаваемых по каналам передачи данных.
	8	Обеспечение управления ВУ через веб-интерфейс при локальном Ethernet подключении.

В случае выхода из строя электрической цепи, к которой подсоединено ВУ, обеспечение энергией выполняет источник резервного питания. В подобной ситуации функционирование ВУ не нарушается, но максимально возможное время работы ВУ зависит от ёмкости источника и количества потребляемой им энергии. Стоимость ВУ рассчитывается комплексно, учитывается микроконтроллер со всем множеством компонент. Таким образом, предпочтение следует отдавать ВУ, которое удовлетворяет всем функциональным требованиям и при этом оптимально по соотношению итоговая стоимость / энергоэффективность.

Предполагается, что устройство будет располагаться в непосредственной близости от входа в помещение. ВУ должно обладать соответствующими размерами, позволяющими разместить его на малой площади. Наименьшей поверхностью в данной ситуации обладает дверь. Таким образом, при встраивании в дверь ВУ должно быть не толще трех сантиметров, при условии, что наиболее распространенная толщина двери четыре сантиметра. Отметим, что толщиной менее трех сантиметров обладает большая часть популярных микроконтроллеров.

Таким образом, нефункциональные требования могут быть сведены в единую таблицу (таблица 3).

Таблица 3. Нефункциональные требования к разрабатываемому ВУ

Нефункциональные требования	Описание
К энергоэффективности	Обеспечение функционирования ВУ в условиях выхода из строя электрической цепи, к которой подсоединено ВУ, за счет энергии резервного источника питания.
К стоимости	Минимизация стоимости микроконтроллера, расширений микроконтроллера, сенсоров и периферии, необходимых для соответствия функциональным требованиям.
К занимаемому пространству	Минимизация размеров микроконтроллера, расширений микроконтроллера, сенсоров и периферии, таким образом, чтобы толщина ВУ не превышала толщины двери.

5. Применение методики для выбора компонентов прототипа системы охраны периметра. В соответствии с функциональными требованиями (таблица 2), каждая из рассматриваемых альтернатив должна поддерживать взаимодействие с внешними электронными компонентами: механическими замками, сканерами бесконтактных карт технологии RFID, инфракрасными датчиками движения, устройствами вывода текстовой, звуковой информации, звуковых и световых сигналов. Набор внешних электронных компонент с их показателями по цене и потреблению электроэнергии сведен в единую таблицу (таблица 4).

Таблица 4. Набор внешних электронных компонентов

Внешний электронный компонент	Выбранное физическое устройство	Потребление (мАм/час)	Стоимость (руб.)
Механический замок	TowerPro SG90	550 [15] (в момент открытия двери)	792
Сканер бесконтактных карт технологии RFID	Grove 125KHz RFID Reader	50 [16]	1296
Инфракрасный датчик движения	**PIR Motion Sensor HC-SR501**	0,05 [17]	216
Устройство вывода текстовой информации	DC 5V Character LCD 16x2	100 [18] (при поднесении бесконтактной карты к сканеру)	1080
Устройство вывода звуковых сигналов	DC 12mA 5V 12mm Piezo Alarm Buzzer	12 [17] (в момент подачи сигнала)	216
Устройство вывода световых сигналов	RGB Light-emitting Diode	20 [19]	7.2
Итого		70,05 – 732,05 (среднее – 200)	2880

Примем, что набор внешних электронных компонент будет единым, поэтому в методике проектирования (раздел 2) не будет осуществляться поиск альтернатив для каждого из внешних электронных компонентов. Влияние набора (таблица 4) внешних электронных компонентов на нефункциональные требования (таблица 3) будет учтено при принятии оптимального с точки зрения нефункциональных требований решения. С учетом функциональных требований, на выходе методики были сформированы альтернативы, представленные в таблице 5.

Таблица 5. Альтернативы, выбранные при помощи методики проектирования безопасных ВУ

№	Набор компонентов защиты (альтернатива)	Описание
1	Arduino Yun, microSD 512 MB	Внутренняя память Arduino Yun ограничена 8 MB, чего недостаточно для соответствия функциональным требованиям к программному обеспечению. Внутреннюю память Arduino Yun можно расширить с помощью microSD.
2	Raspberry Pi B+, Wi-Pi модуль	Raspberry Pi B+ не поддерживает передачу данных по беспроводному каналу передачи данных, а потому не соответствует одному из функциональных требований к аппаратному обеспечению. Wi-Pi модуль разработан специально для Raspberry Pi, чтобы нивелировать данный недостаток.
3	Beaglebone Black, Compact USB Wi-Fi Adapter	Beaglebone Black не поддерживает передачу данных по беспроводному каналу передачи данных, а потому не соответствует одному из функциональных требований к аппаратному обеспечению. Compact USB Wi-Fi Adapter разработан специально для Beaglebone Black, чтобы нивелировать данный недостаток.
4	Intel Galileo Gen 2P Board, Intel Galileo Wi-Fi Kit	Intel Galileo Gen 2P Board не поддерживает передачу данных по беспроводному каналу передачи данных, а потому не соответствует одному из функциональных требований к аппаратному обеспечению. Intel Galileo Wi-Fi Kit разработан специально для Intel Galileo Gen 2P Board, чтобы нивелировать данный недостаток.

В процессе выполнения методики проектирования в выходной набор альтернатив не прошли: микроконтроллер Arduino Mega из-за отсутствия поддержки запуска приложений, написанных на JAVA, Python, C++; Raspberry Pi A+ из-за отсутствия поддержки передачи

данных через Ethernet. Кроме того, Raspberry Pi 2 не был включен в перечень микроконтроллеров, подаваемых на вход методике, так как Raspberry Pi 2 является более дорогим аналогом Raspberry Pi B+, а потому хуже по нефункциональным требованиям.

При анализе энергоэффективности отдельных компонентов и устройств использовались как источники из сети Интернет, так и эксперименты с реальным оборудованием.

Потребление электроэнергии альтернативы 1 включает энергопотребление Arduino Yun и набора внешних электронных компонент. Энергопотребление микроконтроллера Arduino Yun, в свою очередь, зависит от степени нагрузки процессора и может варьироваться между 200 и 300 мАч [21]. Таким образом, потребление электроэнергии альтернативы 1 составляет 400-500 мАч.

Потребление электроэнергии альтернативы 2 включает энергопотребление Raspberry Pi B+, Wi-Pi модуля и набора внешних электронных компонент. Минимальное энергопотребление Raspberry Pi B+ без подключенных внешних устройств составляет 600 мАч [21]. Потребление Wi-Pi модуля составляет 450 мАч [22]. Таким образом, потребление электроэнергии альтернативы 2 составляет 1250 мАч.

Потребление электроэнергии альтернативы 3 включает энергопотребление Beaglebone Black, Compact USB Wi-Fi Adapter и набора внешних электронных компонент. Энергопотребление Beaglebone Black составляет 210-460 мАч [23]. Потребление Compact USB Wi-Fi Adapter составляет 120 мАч (получено экспериментально). Таким образом, потребление электроэнергии альтернативы 3 составляет 780 мАч.

Потребление электроэнергии альтернативы 4 включает энергопотребление Intel Galileo Gen 2P Board, Intel Galileo Wi-Fi Kit и набора внешних электронных компонент. Минимальное энергопотребление Intel Galileo Gen 2P Board без подключенных внешних устройств составляет 800 мАч [24]. Потребление Intel Galileo Wi-Fi Kit составляет около 400мАч [17]. Следовательно, потребление электроэнергии альтернативы 4 составляет 1400 мАч.

Стоимости микроконтроллеров и специфичных для них элементов сформированы исходя из официальных предложений разработчиков и их дистрибьюторов. Более дешевые аналоги (реплики) не рассматривались, так как невозможно гарантировать их совместимость с общим набором внешних компонент. Стоимость неспецифичных компонент сформирована исходя из предложений электронного магазина Amazon [20]. Стоимость внешних компонентов, включая механический замок, сканер бесконтактных карт на технологии RFID, инфра-

красный датчик движения, устройства вывода текстовой информации, звуковых и световых сигналов, составляет 2880 руб. (таблица 6).

Рассмотрим особенности предлагаемых альтернатив, которые также влияют на общую цену:

– В стоимость альтернативы 1 входят: Arduino Yun — 3744 руб. [14], Micro SD карта — 144 руб. [17]. Итоговая стоимость с учтом внешних электронных компонент — 6768 руб.

– В стоимость альтернативы 2 входят: Raspberry Pi B+ — 656 руб. [25], Wi-Pi модуль — 792 руб. [17]. Итоговая стоимость с учетом внешних электронных компонент — 5328 руб.

– В стоимость альтернативы 3 входят: Beaglebone Black — 3600 руб. [26], Compact USB Wi-Fi Adapter — 792 руб. [17]. Итоговая стоимость с учетом внешних электронных компонент — 7272 руб.

– В стоимость альтернативы 4 входят: Intel Galileo Gen 2P Board — 4464 руб. [27], Intel Galileo Wi-Fi Kit — 3240 руб. [17]. Итоговая стоимость с учетом внешних электронных компонент — 10584 руб.

Таблица 6. Сравнение альтернатив по нефункциональным требованиям

Набор компонентов защиты	Потребление энергии (мАч)	Стоимость (₽)	Размер (мм)
Arduino Yun, microSD 512 MB	500	6768	73*53*8
Raspberry Pi B+, Wi-Pi модуль	1250	5328	60*36*7
Beaglebone Black, Compact USB Wi-Fi Adapter	780	7272	86*53*7
Intel Galileo Gen 2P Board, Intel Galileo Wi-Fi Kit	1400	10584	123*72*9

Размеры ВУ напрямую зависят от размера их самой большой части микроконтроллеров. Толщина всех микроконтроллеров соответствует ограничению в 3 сантиметра (при непосредственном размещении в двери). Исходя из нефункциональных требований, были получены результаты, представленные в таблице 6.

Альтернатива 1 показывает лучшую энергоэффективность при средней стоимости. Альтернатива 2 имеет малую энергоэффективность, однако ее стоимость значительно ниже приведенных аналогов, что дает основания для рассмотрения и этого варианта. Альтернатива 3 имеет энергоэффективность и стоимость, близкую к альтернативе 1, но все же несколько уступает ей. Таким образом, альтернативу 3 можно

далее не рассматривать. Альтернатива 4 является наиболее дорогим и наименее энергоэффективным решением в сравнении с остальными наборами компонентов защиты. Дальнейшее рассмотрение альтернативы 4 нецелесообразно.

Требование энергоэффективности подразумевает, что ВУ способно поддерживать функционирование в условиях выхода из строя электрической цепи, к которой подсоединено ВУ, за счет энергии резервного источника питания. Время работы от резервного источника питания для различных альтернатив зависит и от емкости источника резервного питания. Для работы альтернатив 1 и 2 на протяжении 24 часов необходимы аккумуляторы со следующими емкостями: альтернатива 1 — 12000 mAh, альтернатива 2 — 30000 mAh. Необходимая емкость была рассчитана на основе полученных ранее значений среднего энергопотребления.

В качестве источников резервного питания рассматриваются power bank, так как их размер соответствует нефункциональным требованиям. Стоимость power bank зависит от ёмкости и количества поддерживаемых циклов перезарядки и равна 1280 руб. и 2560 руб. для ёмкостей в 12000 mAh и 30000 mAh. Таким образом, итоговая стоимость эксплуатации альтернативы 2 составляет 7888 руб., если учитывать стоимость источника резервного питания. Это значительно выше, чем итоговая стоимость альтернативы 1, которая составляет 8048 руб.

С учетом вышесказанного, оптимальным набором компонентов защиты является альтернатива 1. Итоговый набор компонент защиты: Arduino Yun, microSD 512 MB, TowerPro SG90, Grove 125KHz RFID Reader, PIR Motion Sensor HC-SR501, DC 5V Character LCD 16x2, DC 12mA 5V 12mm Piezo Alarm Buzzer, RGB Light-emitting Diode, power bank 12000 mAh; стоимостью 6768 руб. и энергопотреблением 500mAh.

6. Описание прототипа спроектированной системы охраны периметра. Архитектура системы представлена на рисунке 1:

(1) микроконтроллер Arduino Yun состоит из двух частей:

– процессор ATmega 32U4 (выполняет требования 2, 3 из табл. II), управляющий через sketch [28] сервоприводом, сканнером бесконтактных карт технологии RFID, текстовым экраном, инфракрасным датчиком движения, компонентами вывода световых и звуковых сигналов;

– процессор AR9331 под управлением Linux (выполняет требования 5, 6 из табл. II), содержащий Arduino_Client (выполняет требования 4, 7 из табл. II), который выполняет роль посредника между ATmega 32U4 и Access_App_Server, а также журналирует вход и выход пользователей из помещения;

(2) Access_App_Server предоставляет удаленный доступ к Access_DB_Server для Arduino_Client и для Admin_Client, а также журналирует действия Admin_Client;

(3) база данных Access_DB_Server содержит информацию о пользователях системы, бесконтактных картах, ролях пользователей, устройствах и правах доступа к помещениям для каждой из ролей;

(4) Admin_Client осуществляет управление информацией, хранящейся в базе данных Access_DB_Server, а также журналирование действий администратора;

(5) User_Client журналирует начало и завершение сеанса пользователя с операционной системой;

(6) Syslog_App_Server предоставляет удаленный доступ к Syslog_DB_Server, а также осуществляет генерацию инцидентов безопасности на основе корреляции журналов;

(7) база данных Syslog_DB_Server содержит информацию о событиях и инцидентах безопасности, происходящих в системе.

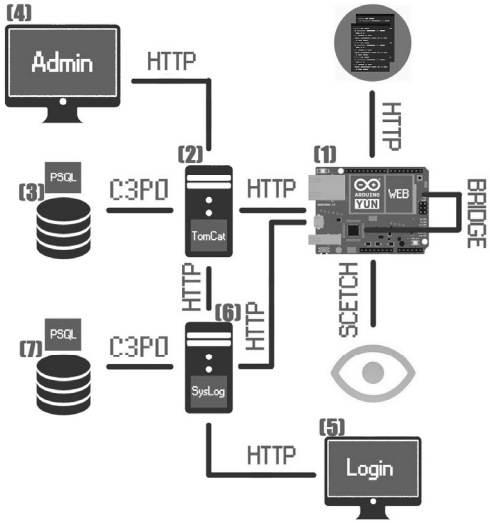

Рис. 1. Архитектура системы

В процессор ATmega 32U4 (1) загружается код, который называется sketch [28]. Любой sketch можно условно разделить на три части: блок инициализации, блок прошивки, блок исполнения. В блоке инициализации подключаются необходимые для работы sketch-библиотеки, объявляются глобальные переменные, задаются специальные обозначения для цифровых или аналоговых PIN. В блоке про-

шивки (setup(){}) осуществляется настройка Arduino, ее подготовка для выполнения функционала, необходимого в блоке исполнения. Блок исполнения (loop(){}) циклически выполняет записанный в нем код, представляя собой алгоритмическое ядро sketch.

Взаимосвязь между ATmega 32U4 и AR9331 Linux-процессорами Arduino Yun осуществляется при помощи библиотек Bridge.h и FileIO.h [28]. Библиотека Bridge.h позволяет запускать shell-команды прямо из sketch, а FileIO.h дает возможность считывать и записывать файлы, принадлежащие файловой системе Linux.

Запуск Arduino_Client осуществляется из sketch благодаря библиотеке Bridge.h посредством выполнения асинхронной (без ожидания завершения) shell-команды. Дальнейшее взаимодействие между sketch и Arduino_Client строится на обработке (чтение/запись) специальных текстовых (.txt) файлов. Это происходит относительно быстро благодаря использованию библиотеки FileIO.h. Запись осуществляется в специальном формате, напоминающем HTTP.

Arduino_Client представлен jar-приложением [29]. Работу jar-приложения клиента можно разделить на два этапа: инициализация и функционирование. При инициализации происходит проверка доступности сервера (2), настройка параметров взаимодействия между AR9331 и ATmega, а также репликация данных администраторов из Access_DB_Server в локальную базу данных и из нее в кэш. Локальная база данных хранит реплицированные данные администраторов. Она необходима на случай, если Access_App_Server станет недоступен. На этапе функционирования предусмотрено разбиение главного потока на прикладные. Каждый поток обрабатывает запросы, приходящие от Arduino, с некоторой периодичностью. В данный момент реализовано четыре потока: первый обрабатывает запросы на получение доступа в помещение, второй проверяет соединение с сервером, третий обеспечивает журналирование, четвертый обновляет локальную БД с некоторой периодичностью или по запросу от ATmega.

При помощи sketch также реализована Arduino_Web_Pannel веб-страница, позволяющая просматривать локальные журналы устройства и инициировать обновления локальной базы данных. Для получения доступа к веб-странице необходимо прямое подключение к микроконтроллеру через Ethernet-кабель, а также прохождение процедуры аутентификации.

Серверная часть (2) и (6) представлена каталогом сервлетов Tomcat [30], на котором развернуто war-приложение. War-приложение разрабатывается в рамках фреймворка Spring [31], который уже имеет готовые решения в области доступа к базе данных (пакет DAO

Support) и реализации базовых требований к безопасности (пакет Spring Security).

Посредником между war-приложением и базами данных (3) и (7) является пул соединений C3P0 [32], который основан на JDBC [29]. C3P0 обеспечивает большую гибкость соединения: он позволяет распределять подключения к базе данных для разных пользователей с разными правами, а также реализует механизмы управления нагрузкой.

База данных Access_DB_Server (3) содержит информацию о пользователях системы, бесконтактных картах, ролях пользователей, устройствах и правах доступа на эти устройства для каждой из ролей.

Клиентская часть администратора (4) представлена jar-приложением, запуск которого осуществляет администратор, предварительно пройдя процедуру аутентификации путем ввода логина и пароля.

Клиентская часть, расположенная на пользовательском компьютере (5), представлена jar-приложением, запуск которого осуществляет операционная система при успешной попытке входа/выхода в/из системы. При запуске клиент (5) отправляет сообщение на Syslog_App_Server (6) с информацией о параметрах входа в систему. При отсутствии сети (как правило, сеть появляется через несколько секунд после входа в систему) клиент пытается отправить сообщение с определенным интервалом вплоть до успешной отправки или выхода из системы.

База данных Syslog_DB_Server (7) содержит информацию о событиях и инцидентах безопасности, происходящих в системе.

Взаимодействие компонентов системы осуществляется в рамках сетевого соединения при помощи HTTP-сообщений, так как сообщения обладают большей надежностью, легко поддаются регулированию, поддерживают асинхронное взаимодействие и обеспечивают легкую интеграцию. Сообщения формируются по принципу построения GET-запросов, так как подобная структура устойчива к ошибкам и обеспечивает легкость интеграции компонентов.

Скорость отклика системы контроля доступа в помещение зависит от скорости работы отдельных компонент ВУ. Так, скорость работы ATmega 32U4 зависит от скорости чтения бесконтактных карт технологии RFID и скорости вывода текстовой информации, составляя 30,1 мс. Скорость взаимодействия между ATmega 32U4 и AR9331 зависит от скорости чтения текстовых файлов и записи в них, составляя в среднем 316,2 мс. Скорость обработки запросов и ответов Access_App_Server составляет 47,1 мс и также может варьироваться в зависимости от степени загруженности сервера или канала связи. Таким образом, среднее время между моментом считывания RFID карты и выводом устройством информации составляет 393,4 мс.

7. Анализ. Методика проектирования защищенных встроенных устройств определяет оптимальную конфигурацию защиты, которая соответствует функциональным требованиям на основе нефункциональных требований. Множество альтернатив компонентов защиты задается в специальной базе данных, которая также содержит информацию о совместимости (наличии или отсутствии возможных конфликтов) между компонентами защиты. Оптимальное решение, получаемое на основе работы методики, напрямую зависит от содержимого специальной базы данных, а также от заданных функциональных и нефункциональных требований. Таким образом, качество предоставляемого методикой набора зависит от полноты и актуальности специальной базы данных и корректности заданных требований. Поэтому в общем случае методика проектирования защищенных встроенных устройств не является полноценной заменой экспертного мнения.

Эксперт в области компонентов защиты встроенных устройств, обладая знаниями о специализированных решениях, с большой долей вероятности подберет достаточно эффективный набор компонентов защиты. Но результат работы методики способен принести пользу в случае значительного числа функциональных требований защиты и рассматриваемых альтернатив компонентов защиты в условиях использования программного средства Конфигуратор. Данное средство позволяет автоматизировать стадии 6 и 8 методики, ручное выполнение которых затруднительно. Конфигуратор также может предложить альтернативы субъективным предпочтениям и известным решениям эксперта.

К особенностям предложенной методики можно отнести и то, что она предоставляет возможность заменить собой функции эксперта, определяющего выбор необходимых компонентов защиты, базируясь на заложенных экспертных знаниях, позволяя разработчикам системы сфокусироваться непосредственно на разработке системы защиты и тем самым упростить сложность процесса разработки.

В соответствии с [33] предлагаемая методика относится к категории технических процессов и позволяет реализовывать организационные и проектные функции для оптимизации пользы и снижения рисков, являющихся следствием технических решений и действий [33]. В отличие от [33] предлагаемая методика определяет процесс проектирования защищенных встроенных устройств в части анализа системных требований, проектирования архитектуры системы, процесса реализации с учетом нефункциональных характеристик, входящих в состав компонентов защиты с решением поставленной оптимизационной задачи и с использованием предложенной эвристики.

Отметим также, что оптимальность получаемого в результате применения методики решения понимается в терминах достижения наилучших значений заданных нефункциональных показателей при фиксированных ограничениях на параметры защищенности. Поэтому задача проверки выполнимости критериев оценки информационной безопасности [34] выходит за рамки настоящей статьи. Тогда как точность решения задачи дискретной оптимизации на множестве конфигураций защиты определяется точностью исходных значений нефункциональных показателей компонентов защиты.

В соответствии с [35] предложенную методику можно отнести к следующим стадиям и этапам создания автоматизированных систем в защищенном исполнении [35]: формирование требований к системе защиты информации, разработка (проектирование) системы защиты информации. В соответствии с [36] проектируемую систему охраны периметра можно отнести в качестве интегрированной системы безопасности – подсистемы комплексной системы безопасности в части реализации функций охранной и пожарной сигнализации [36] с учетом требований к энергоэффективности, стоимости и физическим параметрам отдельных компонентов защиты.

На примере разработанного прототипа системы охраны периметра помещения представлена предлагаемая методика, ее основные стадии, а также способы отбора и оценки возможных компонентов защиты на предмет получения оптимального решения. При этом отметим, что задача разработки полнофункционального программно-технического решения по организации системы охраны периметра и контроля доступа коммерческого или ведомственного уровня с учетом всех актуальных требований и критериев из приведенных выше стандартов в предметной области выходит за рамки проведенного исследования. При этом такие требования и критерии, будучи специфицированы в терминах входных данных методики, могут быть реализованы посредством применения методики путем задания соответствующих функциональных требований к защите.

8. Заключение. В работе предложена методика проектирования защищенных встроенных устройств на основе комбинирования компонентов защиты. Особенностью методики является использование правил выбора компонентов защиты с учетом функциональных и нефункциональных характеристик компонентов защиты, ограничений устройства и связей между компонентами с использованием оптимизационного подхода. Для проверки методики была выбрана задача проектирования защищенной системы охраны периметра в части реализации функций контроля доступа в помещение. Данная система

представляет собой встроенное устройство, расположенное в дверях между помещениями и подключенное к механизму управления замком. С помощью разработанной методики был осуществлен выбор наиболее подходящего для решения данной задачи микроконтроллера и дополнительных компонент.

Литература

1. *Vasilevskaya M.* Designing Security-enhanced Embedded Systems: Bridging Two Islands of Expertise // PhD thesis. Linkoping Studies in Science and Technology. Sweden. 2013.
2. *Desnitsky V., Kotenko I., Chechulin A.* Configuration-based approach to embedded device security // Springer-Verlag. 2012. LNCS. 7531. pp. 270–285.
3. *Desnitsky V., Kotenko I.* Expert Knowledge based Design and Verification of Secure Systems with Embedded Devices // Springer-Verlag. 2014. LNCS 8708. pp. 194–210.
4. *Henzinger T., Sifakis J.* The Embedded Systems Design Challenge // Springer-Verlag. LNCS 4085. 2006. pp.1–15.
5. Object Management Group. The UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded Systems. version 1.1. 2011.
6. *Hwang D., Schaumont P., Tiri K., Verbauwhede I.* Securing Embedded Systems // IEEE Security and Privacy. 2006. vol. 4. no. 2. pp. 40–49.
7. *Knezevic M., Rozic V., Verbauwhede I.* Design Methods for Embedded Security // Telfor Journal. 2009. vol. 1. no. 2. pp. 69–72.
8. *Moyers B., Dunning J., Marchany R., Tron J.* Effects of Wi-Fi and Bluetooth Battery Exhaustion Attacks on Mobile Devices // Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS'10). IEEE Computer Society. 2010. pp.1–9.
9. *Gogniat G., Wolf T., Burleson W.* Reconfigurable Security Primitive for Embedded Systems // Proceedings of International Symposium on In System-on-Chip. 2005. pp. 23–28.
10. *Norman J.* Open Source Physical Security. URL: www.layerone.org/wp-content/uploads/2012/07/LayerOne2012-John_Norman-DIY_Access_Control.pdf (дата обращения 29.09.2016).
11. *Nojmol I.* How can Access Control Systems Improve Security and Reduce Costs? // Public Sector Estates Management, September. 2014. 8 p. URL: http://www.assaabloy.co.uk/Other/ASSA/ASSA%20ABLOY/White%20Papers/ASSA_Smartair_Whitepaper%20V3.pdf (дата обращения 29.09.2016).
12. *Ruiz J., Harjani R., Maña A., Desnitsky V., Kotenko I., Chechulin A.* A Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components // Proceedings of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP2012). Munich. Germany. 2012. pp. 261–268.
13. *Rae A., Wildman L.* A Taxonomy of Attacks on Secure Devices // Australian Information Warfare and IT Security. 2003. pp. 251–264.
14. *Abraham D., Dolan G., Double G., Stevens J.* Transaction security system // IBM Systems Journal. 1991. pp. 206–228.
15. Intel Galileo distributor shop. URL: http://newegg.com/ (дата обращения: 20.05.2015).
16. Electronic shop. URL: http://nettigo.pl/ (дата обращения: 20.05.2015).
17. Intel Galileo distributor shop. URL: http://mouser.com/ (дата обращения: 20.05.2015).

18. Electronic shop. URL: http://seeedstudio.com/ (дата обращения: 20.05.2015).
19. Energy efficiency of DC 5V Character LCD 16x2. URL: http://melt.com/ (дата обращения: 20.05.2015).
20. Amazon. URL:http://www.amazon.com (дата обращения: 20.05.2015).
21. Arduino forum. URL: http://forum.arduino.cc/ (дата обращения: 20.05.2015).
22. News portal. URL: http://linuxgizmos.com/ (дата обращения: 20.05.2015).
23. Mark VandeWettering`s blog. URL: http://brainwagon.org/ (дата обращения: 20.05.2015).
24. Beaglebone distributor shop. URL: http://digikey.com/ (дата обращения: 20.05.2015).
25. Arduino shop. URL:http://store.arduino.cc/ (дата обращения: 20.05.2015).
26. Raspberry-pi distributor shop. URL: http://alliedelec.com/ (дата обращения: 20.05.2015).
27. Beaglebone distributor shop. URL: https://adafruit.com/ (дата обращения: 20.05.2015).
28. Ebay. URL: http://ebay.com/ (дата обращения: 20.05.2015).
29. C3p0 pooling. URL: http://mchange.com/projects/c3p0/ (дата обращения: 20.05.2015).
30. Arduino references. URL: http://arduino.cc/en/Reference/ (дата обращения: 20.05.2015).
31. Apache Tomcat. URL: http://tomcat.apache.org/ (дата обращения: 20.05.2015).
32. Spring Framework. URL: http://projects.spring.io/spring-framework/ (дата обращения: 20.05.2015).
33. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств // М.: Госстандарт России. 2010.
34. ГОСТ Р ИСО/МЭК 15408-1-2008. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий // М.: Госстандарт России. 2008.
35. ГОСТ Р 51583-2014. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения // М.: Госстандарт России. 2014.
36. ГОСТ Р 53704-2009. Системы безопасности комплексные и интегрированные. Общие технические требования // М.: Госстандарт России. 2009.

Десницкий Василий Алексеевич — к-т техн. наук, старший научный сотрудник лаборатории проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: интернет вещей, безопасность встроенных устройств, защита ПО, методы формальной верификации. Число научных публикаций — 90. vasily.desnitsky@mail.ru, http://comsec.spb.ru/desnitsky; 14-я линия В.О., 39, ком. 205, Санкт-Петербург, 199178; р.т.: +7(812)328-71-81.

Чечулин Андрей Алексеевич — к-т техн. наук, старший научный сотрудник лаборатории проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, обнаружение вторжений, анализ сетевого трафика, анализ уязвимостей. Число научных публикаций — 150. andreych@bk.ru, http://comsec.spb.ru/ru/staff/chechulin; 14-я линия В.О., 39, ком. 205, Санкт-Петербург, 199178; р.т.: +7(812)328-71-81.

Котенко Игорь Витальевич — д-р техн. наук, профессор, заведующий лабораторией проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, в том числе управление политиками безопасности, разграничение доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер–терроризму. Число научных публикаций — 450. ivkote@comsec.spb.ru, http://www.comsec.spb.ru; 14-я линия В.О., 39, Санкт-Петербург, 199178; р.т.: +7-(812)-328-71-81, Факс: +7(812)328–4450.

Левшун Дмитрий Сергеевич — программист лаборатории проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН), студент, Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В.И. Ульянова (Ленина). Область научных интересов: компьютерная безопасность, защита встроенных устройств, системы киберфизической безопасности, безопасность распределённых систем, корреляция событий безопасности. Число научных публикаций — 5. levshun@comsec.spb.ru, http://comsec.spb.ru/levshun; 14-я линия В.О., 39, ком. 205, Санкт-Петербург, 199178; р.т.: +7-(812)-328-71-81.

Коломеец Максим Вадимович — программист лаборатории проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН), студент, Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В.И. Ульянова (Ленина). Область научных интересов: безопасность распределенных систем, визуализация данных. Число научных публикаций — 10. guardeecwalker@gmail.com; 14-я линия В.О., д. 39, ком. 205, Санкт-Петербург, 199178; р.т.: +7(812)328-71-81.

Поддержка исследований. Работа выполнена при частичной финансовой поддержке РФФИ (проекты №14-07-00697, 14-07-00417, 15-07-07451, 16-37-00338, 16-29-09482 офи_м, 16-37-50035), при частичной поддержке бюджетных тем № 0073-2015-0004 и 0073-2015-0007, а также гранта РНФ 15-11-30029 в СПИИРАН.

V.A. DESNITSKY, A.A. CHECHULIN, I.V. KOTENKO, D.S. LEVSHUN, M.V. KOLOMEEC

COMBINED DESIGN TECHNIQUE FOR SECURE EMBEDDED DEVICES EXEMPLIFIED BY A PERIMETER PROTECTION SYSTEM

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.**

Abstract. In terms of information security, embedded devices are elements of complex cyber-physical systems, systems of the Internet of Things, working in a potentially hostile environment. Therefore, the development of such devices is a challenging problem, often requiring expert solutions. The complexity of developing secure embedded devices is due to different types of potential threats and attacks to the device, as well as the fact that in practice security of embedded devices is usually considered in the final stages of the development process in the form of adding additional security features. In the paper, we propose a design technique aimed at the development of safe and energy-efficient cyber-physical and embedded devices. This technique organizes a search for the best combination of security components on the basis of solving an optimization problem. The efficiency of the proposed technique is demonstrated through the development of a secure system to protect a room perimeter.

Keywords: Embedded devices, cyber-physical systems, perimeter control, design of secure cyber-physical systems.

Desnitsky Vasily Alekseevich — Ph.D., senior researcher of computer security problems laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science (SPIIRAS). Research interests: Internet of Things, embedded security, software protection, methods of formal verification. The number of publications — 90. vasily.desnitsky@mail.ru, http://comsec.spb.ru/desnitsky; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7(812)328-71-81.

Chechulin Andrey Alexeevich — Ph.D., senior researcher of computer security problems laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science (SPIIRAS). Research interests: computer network security, intrusion detection, analysis of the network traffic, vulnerability analysis. The number of publications — 150. andreych@bk.ru, http://comsec.spb.ru/ru/staff/chechulin; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7(812)328-71-81.

Kotenko Igor Vitalievich — Ph.D., Dr. Sci., professor, head of computer security problems Laboratory, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS). Research interests: computer network security, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, malware protection, verification of security systems, digital right management, modeling, simulation and visualization technologies for counteraction to cyber terrorism. The number of publications — 450. ivkote@comsec.spb.ru, http://www.comsec.spb.ru; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7-(812)-328-71-81, Fax: +7(812)328–4450.

Levshun Dmitry Sergeevich — software developer of computer security problems laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science (SPIIRAS), student, Saint Petersburg Electrotechnical University "LETI". Research interests:

distributed system security, embedded devices, event correlation, cyber-physical security systems. The number of publications — 5. levshun@comsec.spb.ru, http://comsec.spb.ru/levshun; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7-(812)-328-71-81.

Kolomeec Maxim Vadimovich — developer of computer security problems laboratory, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS), student, Saint Petersburg Electrotechnical University "LETI". Research interests: distributed system security, security visualization. The number of publications — 10. guard-eecwalker@gmail.com; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7(812)328-71-81.

Acknowledgements. This research was partially financially supported by grants of RFBR (projects No. 14-07-00697, 14-07-00417, 15-07-07451, 16-37-00338, 16-29-09482 ofi_m, 16-37-50035), by the budget (projects No. 0073-2015-0004 and 0073-2015-0007) as well as by Grant of Russian Science Foundation No. 15-11-30029 in SPIIRAS.

References

1. Vasilevskaya M. Designing Security-enhanced Embedded Systems: Bridging Two Islands of Expertise. PhD thesis. Linkoping Studies in Science and Technology. Sweden. 2013.
2. Desnitsky V., Kotenko I., Chechulin A. Configuration-based approach to embedded device security. Springer-Verlag. 2012. LNCS 7531. pp. 270–285.
3. Desnitsky V., Kotenko I. Expert Knowledge based Design and Verification of Secure Systems with Embedded Devices. Springer-Verlag. LNCS 8708. 2014. pp. 194–210.
4. Henzinger T., Sifakis J. The Embedded Systems Design Challenge. Springer-Verlag. 2006. LNCS 4085. pp.1–15.
5. Object Management Group. The UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded Systems. version 1.1. 2011.
6. Hwang D., Schaumont P., Tiri K., Verbauwhede I. Securing Embedded Systems. *IEEE Security and Privacy*. 2006. vol. 4. no. 2. pp.40–49.
7. Knezevic M., Rozic V., Verbauwhede I. Design Methods for Embedded Security. *Telfor Journal*. 2009. vol. 1. no. 2. pp. 69–72.
8. Moyers B., Dunning J., Marchany R., Tron J. Effects of Wi-Fi and Bluetooth Battery Exhaustion Attacks on Mobile Devices. Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS'10). IEEE Computer Society. 2010. pp. 1–9.
9. Gogniat G., Wolf T., Burleson W. *Reconfigurable Security Primitive for Embedded Systems*. Proceedings of International Symposium on In System-on-Chip. 2005. pp. 23–28.
10. Norman J. Open Source Physical Security. July 2012. URL: www.layerone.org/wp-content/uploads/2012/07/LayerOne2012-John_Norman-DIY_Access_Control.pdf (accessed 29.09.2016).
11. Nojmol I. How can Access Control Systems Improve Security and Reduce Costs? Public Sector Estates Management, September. 2014. 8 p. URL: http://www.assaabloy.co.uk/Other/ASSA/ASSA%20ABLOY/White%20Papers/ASSA_Smartair_Whitepaper%20V3.pdf (accessed 29.09.2016).
12. Ruiz J., Harjani R., Maña A., Desnitsky V., Kotenko I., Chechulin A. A Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components. Proceedings of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP2012). Munich. Germany. 2012. pp. 261–268.

13. Rae A., Wildman L. A Taxonomy of Attacks on Secure Devices. Australian Information Warfare and IT Security. 2003. pp.251–264.
14. Abraham D., Dolan G., Double G., Stevens J. Transaction security system // *IBM Systems Journal*. 1991. pp.206–228.
15. Intel Galileo distributor shop. Available at: http://newegg.com/ (accessed 20.05.2015).
16. Electronic shop. Available at: http://nettigo.pl/ (accessed 20.05.2015).
17. Intel Galileo distributor shop. Available at: http://mouser.com/ (accessed 20.05.2015).
18. Electronic shop. Available at: http://seeedstudio.com/ (accessed 20.05.2015).
19. Energy efficiency of DC 5V Character LCD 16x2. Available at: http://melt.com/ (accessed 20.05.2015).
20. Amazon Available at: http://www.amazon.com (accessed 20.05.2015).
21. Arduino forum. Available at: http://forum.arduino.cc/ (accessed 20.05.2015).
22. News portal. Available at: http://linuxgizmos.com/ (accessed 20.05.2015).
23. Mark VandeWettering`s blog. Available at: http://brainwagon.org/ (accessed 20.05.2015).
24. Beaglebone distributor shop. Available at: http://digikey.com/ (accessed 20.05.2015).
25. Arduino shop. Available at: http://store.arduino.cc/ (accessed 20.05.2015).
26. Raspberry-pi distributor shop. Available at: http://alliedelec.com/ (accessed 20.05.2015).
27. Beaglebone distributor shop. Available at: https://adafruit.com/ (accessed 20.05.2015).
28. Ebay. Available at: http://ebay.com/ (accessed 20.05.2015).
29. C3p0 pooling. Available at: http://mchange.com/projects/c3p0/ (accessed 20.05.2015).
30. Arduino references. Available at: http://arduino.cc/en/Reference/ (accessed 20.05.2015).
31. Apache Tomcat. Available at: http://tomcat.apache.org/ (accessed 20.05.2015).
32. Spring Framework. Available at: http://projects.spring.io/spring-framework/ (accessed 20.05.2015).
33. GOST R ISO/MJK 12207-2010. [Information technology. System and software engineering. The processes of software life cycle]. M.: Gosstandart Rossii. 2010. (In Russ.).
34. GOST R ISO/MJK 15408-1-2008. [Information technology. Methods and security features. Criteria for Information Technology Security Evaluation]. M.: Gosstandart Rossii. 2008. (In Russ.).
35. GOST R 51583-2014. [Data protection. The order of creation of automated systems in the protected design. General provisions]. M.: Gosstandart Rossii. 2014. (In Russ.).
36. GOST R 53704-2009. [Security systems complex and integrated. General specifications]. 2009. (In Russ.).