Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий

Работа допущена к защите
Руководитель ОП
_____ А.В. Щукин
«_____» _____ 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
РАБОТА БАКАЛАВРА
СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ЛЕГКОВЕСНОЙ КРИПТОГРАФИИ ДЛЯ УСТРОЙСТВ ИНТЕРНЕТА ВЕЩЕЙ

по направлению подготовки 09.03.03 Прикладная информатика
Направленность (профиль) 09.03.03_03 Прикладная информатика в области информационных ресурсов

Выполнил
студент гр. 3530903/60301 Д.М. Момот

Руководитель
доцент ВШИСиСТ,
к. т. н., доцент А.В. Сергеев

Консультант
по нормоконтролю В.А. Пархоменко

Санкт-Петербург
2020

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и технологий

УТВЕРЖДАЮ
Руководитель ОП
_____ А.В. Щукин
«_____» _____ 2020г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Момоту Даниэлю Михайловичу гр. 3530903/60301

1. Тема работы: Сравнительный анализ алгоритмов легковесной криптографии для устройств интернета вещей.
2. Срок сдачи студентом законченной работы: 20.05.2020.
3. Исходные данные по работе: спецификация языка C++11 [3.2], библиотеки языка C++ [3.1].
   3.1. C++ Standard Library headers. — URL: https://en.cppreference.com/w/cpp/header (visited on 10.06.2020).
   3.2. Programming Languages — C++: Standard. — 2013. — URL: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3690.pdf.
4. Содержание работы (перечень подлежащих разработке вопросов):
   4.1. Обзор теоретических источников по темам легковесной криптографии и интернета вещей.
   4.2. Обзор существующих стандартов и технических решений.
   4.3. Анализ различных видов алгоритмов.
   4.4. Реализация избранных алгоритмов.
   4.5. Ранжирование видов алгоритмов по степени пригодности к использованию в устройствах IoT. Формулировка рекомендаций по их использованию.
5. Перечень графического материала (с указанием обязательных чертежей):
   5.1. Графики, отражающие результаты эксперимента.
6. Консультанты по работе:

6.1. Ассистент ВШИСиСТ, В.А. Пархоменко (нормоконтроль).

7. Дата выдачи задания: 03.02.2020.

Руководитель ВКР _____ А.В. Сергеев

Задание принял к исполнению 03.02.2020

Студент _____ Д.М. Момот

РЕФЕРАТ

На 38 с., 4 рисунка, 6 таблиц, 2 приложения.

КЛЮЧЕВЫЕ СЛОВА: СТИЛЕВОЕ ОФОРМЛЕНИЕ САЙТА, УПРАВЛЕНИЕ КОНТЕНТОМ, PHP, MYSQL, АРХИТЕКТУРА СИСТЕМЫ.[1]

Тема выпускной квалификационной работы: «Сравнительный анализ алгоритмов легковесной криптографии для устройств интернета вещей»[2].

В данной работе изложена сущность подхода к созданию динамического информационного портала на основе использования открытых технологий Apache, MySQL и PHP. Даны общие понятия и классификация IT-систем такого класса. Проведен анализ систем-прототипов. Изучена технология создания указанного класса информационных систем. Разработана конкретная программная реализация динамического информационного портала на примере портала выбранной тематики...[3]

В данной работе изложена сущность подхода к созданию динамического информационного портала на основе использования открытых технологий Apache, MySQL и PHP. Даны общие понятия и классификация IT-систем такого класса. Проведен анализ систем-прототипов. Изучена технология создания указанного класса информационных систем. Разработана конкретная программная реализация динамического информационного портала на примере портала выбранной тематики...

ABSTRACT

38 p., 4 figures, 6 tables, 2 appendices.

KEYWORDS: STYLE REGISTRATION, CONTENT MANAGEMENT, PHP, MYSQL, SYSTEM ARCHITECTURE.

---

[1]Всего **слов**: от 3 до 15. Всего **слов и словосочетаний**: от 3 до 5. Оформляются в именительном падеже множественного числа (или в единственном числе, если нет другой формы), оформленных по правилам русского языка. *Внимание! Размещение сноски после точки является примером как запрещено оформлять сноски.*

[2]Реферат **должен содержать**: предмет, тему, цель ВКР; метод или методологию проведения ВКР: результаты ВКР: область применения результатов ВКР; выводы.

[3]ОТ 1000 ДО 1500 печатных знаков (ГОСТ Р 7.0.99-2018 СИБИД) на русский или английский текст. Текст реферата повторён дважды на русском и английском языке для демонстрации подхода к нумерации страниц.

The subject of the graduate qualification work is «Comparative analysis of lightweight cryptography algorithms for IoT devices».

In the given work the essence of the approach to creation of a dynamic information portal on the basis of use of open technologies Apache, MySQL and PHP is stated. The general concepts and classification of IT-systems of such class are given. The analysis of systems-prototypes is lead. The technology of creation of the specified class of information systems is investigated. Concrete program realization of a dynamic information portal on an example of a portal of the chosen subjects is developed...

In the given work the essence of the approach to creation of a dynamic information portal on the basis of use of open technologies Apache, MySQL and PHP is stated. The general concepts and classification of IT-systems of such class are given. The analysis of systems-prototypes is lead. The technology of creation of the specified class of information systems is investigated. Concrete program realization of a dynamic information portal on an example of a portal of the chosen subjects is developed...

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В последние десятилетия сетевые технологии прочно вошли в жизнь человека. Изначально они были представлены локальными сетями, затем была создана сеть Интернет. В настоящее время одним из бурно развивающихся направлений сетевых технологий является интернет вещей (Internet of Things, IoT). Так, за период 2015-2018 гг. доля IoT-устройств среди всех устройств увеличилась с 27% до 39%, и, согласно прогнозу, достигнет 63% к 2025 году [46].

Интернет вещей – это вычислительная сеть физических предметов (устройств, «вещей»), оснащенных встроенной технологией для взаимодействия друг с другом или с внешней средой [33].

Актуальность исследования. Одной из важных задач при проектировании IoT является обеспечение должного уровня безопасности передаваемых данных. Особенно это важно для медицинских устройств[2]. Уже сейчас правительства развитых стран начинают принимать законы, регламентирующие защиту IoT–устройств[30][9][8]. В то же время фактическая безопасность устройств интернета вещей оставляет желать лучшего. Так, согласно исследованию корпорации HP 2014-го года, 70% устройств IoT передавали данные, в том числе конфиденциального характера, вообще без шифрования[32]! По этой причине изучение и развитие средств защиты IoT-устройств является актуальной задачей.

Безопасность устройств с ограниченными энергетическими ресурсами (а именно такими являются устройства интернета вещей) изучает раздел криптографии, называемый легковесной криптографией (lightweight cryptography, LWC). Также возможны термины «облегченная криптография», «малоресурсная криптография», «низкоэнергетическая криптография». Она рассматривает криптографические алгоритмы в контексте их требовательности к ресурсам устройства и количеству логических элементов, требуемых для реализации алгоритмов. Рассматриваемые ею алгоритмы называются алгоритмами легковесной криптографии (легковесными алгоритмами, LW-алгоритмами). LW-алгоритмы могут иметь программную или аппаратную реализацию.

Объектом исследования являются алгоритмы легковесной криптографии, ориентированные на использование в устройствах интернета вещей.

Предметом исследования являются следующие характеристики легковесных алгоритмов: тип алгоритма, требования к устройству, производительность. Алгоритмы рассматриваются с точки зрения программной реализации.

Целью исследования является сравнение легковесных алгоритмов и определение степени их пригодности к использованию в устройствах IoT. Для достижения выбранной цели поставлены следующие **задачи**:

A. Обзор теоретических источников по темам легковесной криптографии и интернета вещей.
B. Обзор существующих стандартов и технических решений.
C. Анализ различных видов алгоритмов.
D. Реализация избранных алгоритмов.
E. Ранжирование видов алгоритмов по степени пригодности к использованию в устройствах IoT. Формулировка рекомендаций по их использованию.

Гипотеза исследования. Предполагается, что значительное число классических шифров с определенным ослаблением могут использоваться в качестве легковесных алгоритмов. С другой стороны, многие алгоритмы и классы алгоритмов по тем или иным соображениям однозначно не могут быть использованы в качестве LW-алгоритмов. Ряд алгоритмов может быть использован наилучшим образом при определенных условиях. Может быть создана методика, позволяющая приблизительно оценить производительность реализации легковесного алгоритма с использованием только персонального компьютера, без необходимости взаимодействия с низкоресурсными устройствами.

В данной работе используются такие **методы исследования**, как:

– анализ технической литературы;
– изучение существующих спецификаций алгоритмов, стандартов, технологических систем;
– декомпозиция алгоритмов;
– сравнение алгоритмов, подходов к шифрованию, структурных частей алгоритмов;
– реализация алгоритмов на языке C;
– определение методологии их тестирования;
– реализация конкретной методики тестирования;
– анализ результатов тестирования;
– синтез выводов по результатам тестирования;
– обобщение результатов работы.

Данное исследование имеет высокую **практическую значимость**. Результаты исследования могут быть использованы при выборе алгоритма шифрования и

режима его работы при проектировании системы защиты вычислительной сети IoT. Кроме того, работа в данном направлении может быть продолжена: следующим шагом возможна оптимизация существующих легковесных алгоритмов или создание новых LW-алгоритмов. Предложенная методология тестирования времени работы алгоритмов и их энергопотребления может быть использована при тестировании других алгоритмов. Присутствует и **научная значимость**. Можно исследовать «побочные» вопросы, возникшие в процессе выполнения работы.

Введение раскрывает актуальность, определяет степень научной разработки темы, объект, предмет, цель, задачи и методы исследования, раскрывает теоретическую и практическую значимость работы.

В первой главе содержатся краткое теоретическое введение в тему криптографии и интернета вещей, описываются основные угрозы безопасности систем IoT.

Во второй главе вводится понятие легковесных криптографических алгоритмов. Описываются требования к ним и их применение для нейтрализации угроз безопасности системам IoT, оно сравнивается с нейтрализацией угроз в системах общего назначения). Проводится анализ различных видов алгоритмов на пригодность их для такого применения, рассматриваются наиболее популярные их представители.

В третьей главе описывается методология тестирования производительности и энергопотребления легковесных криптоалгоритмов на персональном компьютере.

В четвертой главе содержится информация о реализации алгоритмов в программном коде, способе и порядке тестирования производительности алгоритмов, а также дальнейшей обработки результатов тестирования. Проводится анализ полученных результатов.

В заключении подводятся итоги работы, приводятся краткие перспективы использования результатов работы.

ГЛАВА 1. ОБЗОР ТЕОРЕТИЧЕСКОЙ ЛИТЕРАТУРЫ ПО КРИПТОГРАФИИ И ИНТЕРНЕТУ ВЕЩЕЙ

В данной главе приводится основная терминологеская и теоретическая база, на которой будут строиться дальнейшие главы. Рассматриваются основные понятия криптографии и интернета вещей. Описывается уровневая модель систем интернета вещей, выделяется каждый уровень и атаки на него.

1.1. Криптографические алгоритмы: основные понятия

1.1.1. Информация и криптография

Человеческая цивилизация в течение всего времени существования работает с различной информацией. Информация может представлять собой как исторические сведения или описания технологических процессов, так и частные данные человека или группы лиц. Сведения, составляющие информацию и представляющие определенную ценность, должны быть защищены от лиц, не имеющих соответствующих прав доступа. Поэтому задача защиты информации всегда была актуальной.

Защитить информацию можно тремя способами [14].

A. Создание абсолютно надежного и изолированного от доступа извне хранилища информации и инфраструктуры для него. Это является крайне трудоемкой и дорогостоящей задачей.
B. Скрытие факта существования или передачи информации. Средства и методы такого скрытия изучает стеганография.
C. Хранение и передача информации, преобразованной таким образом, чтобы обратное преобразование могли совершить только определенные лица. Средства и методы такого преобразования информации изучает криптография.

Формализуем вышесказанное.

Информация – сведения, передаваемые одними людьми другим людям устным, письменным или каким-либо другим способом, а также сам процесс передачи или получения этих сведений [10].

Защита информации – деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных

воздействий на защищаемую информацию [4]. Утечка – неконтролируемое распространение защищаемой информации в результате ее разглашения, несанкционированного доступа к информации и получения защищаемой информации иностранными разведками [5].

Криптография -— инженерно-техническая дисциплина, изучающая математические методы защиты информации (шифры). Криптография включает в себя криптосинтез и криптоанализ. Криптосинтез изучает подходы к разработке шифров. Криптоанализ изучает подходы к вскрытию шифров [14].

1.1.2. Шифры: основные понятия и принципы

Итак, шифр (криптографический алгоритм, шифрующая функция) – преобразование данных, обеспечивающее их защищённость (зашифрованность). Это преобразование вида f: x→y, где x – исходные, незашифрованные данные (представленные в виде целого числа), y – полученный зашифрованный текст (также целое число), f – шифрующая функция.

Дешифрующая функция – функция, обратная к шифрующей: $\hat{f}(-1)$: y→x. Результатом применения дешифрующей функции к зашифрованным данным являются исходные данные.

Для того, чтобы шифрование было эффективным, необходимо, чтобы шифрующая функция f была легко вычислимой, для затруднения дешифровки дешифрующая функция $\hat{f}(-1)$ должна быть достаточно сложна для вычисления [12]. Функции такого вида называются односторонними.

Функция f называется *односторонней*, если выполняются условия [12]:

A. для любого x из некоторого множества существует эффективный алгоритм вычисления y=f(x);
B. не существует эффективного алгоритма обращения функции f.

Однако шифр должен быть не только недоступным для взлома злоумышленником. Законный получатель должен иметь возможность эффективно получить исходные данные. Это требование формализуется как «функция с секретом» или «односторонняя функция с секретом».

Функция f называется *односторонней функцией с секретом*, если выполняются условия [12]:

A. для любого x из некоторого множества существует эффективный алгоритм вычисления y=f(x);

В. функция f обладает «секретным свойством» k, таким что:.

1. при использовании свойства k можно построить эффективный алгоритм построения обратной функции f$\hat{}$(-1);
2. если свойство k неизвестно, то не существует эффективного алгоритма обращения функции f.

Итак, при использовании шифра на основе функции с секретом, шифровать (вычислять функцию f) могут все, а эффективно расшифровывать – только лица, которым известно секретное свойство k.

Секретное свойство также называют *ключом шифра*. Он представляет собой число или набор чисел (параметров алгоритма).

Главная часть шифра, его сердце – некоторая *трудная* задана, которая используется для гарантии того, что узнать секретное свойство (т. е. обратить шифрующую функцию и получить доступ к зашифрованным данным) очень трудно. Пара примеров трудных задач:

- Задача факторизации – разложение большого числа на простые множители (используется, например, в алгоритме RSA). В этом случае секретное свойство – факторизуемое число.
- Задача дискретного логарифмирования – обращение функции вида f(x)=a$\hat{}$x mod p, где p – большое простое число, a – параметр, подобранный для конкретного p (используется, например, в схеме Эль-Гамаля).

Однако если будет представлен алгоритм, эффективно решающий трудную задачу (т. е. задача перестанет быть трудной), шифры на ее основе мгновенно устареют. Поэтому поиск трудных задач (и, наоборот, поиски быстро решающих их алгоритмов) является одним из важных направлений криптографии.

С другой стороны, в секрете хранится только ключ алгоритма, сам алгоритм должен быть открытым. То есть вся криптографическая стойкость алгоритма должна содержаться в ключе, а злоумышленник может знать о криптографической схеме все, кроме ключа. Это правило называется *принципом Керкгоффса*. Его выполнение обеспечивает надежность системы: даже в случае подбора ключа (методом перебора или, например, шантажа) злоумышленником достаточно его сменить, и начинать взлом нужно с самого начала. Похожий принцип «враг знает систему» был (вероятно, независимо) сформулирован Шенноном и называется *максимой Шеннона* [18].

Это, однако, является только рекомендацией. В правительственных, военных и ряде других областей применяется противоположный принцип: безопасность

через неясность. Его суть состоит в скрытии внутренней структуры системы безопасности. Для шифров общего назначения он не рекомендуется, их, напротив, публикуют и обсуждают, чтобы коллективно найти уязвимости и возможности улучшения [44].

1.1.3. Симметричная криптография

Изначально существовали только симметричные схемы шифрования. Они предполагают наличие одного ключа, используемого как для шифрования, так и для расшифровки сообщений. Он хранится в тайне, поэтому такие схемы также называются схемами с закрытым ключом. Классический пример – алгоритм DES.

Симметричные шифры в основном состоят из двух главных компонентов: шифры подстановки (перестановка символов/битов сообщения) и шифры замены (замена символов/битов по отдельности на другие по некоторому правилу) [14].

Симметричные шифры делятся на две категории: блочные и потоковые. Потоковые криптоалгоритмы обрабатывают данные побитово или побайтово. Блочные алгоритмы обрабатывают данные целыми группами битов (блоками), обычно размер блока кратен 64 и составляет от 64 до 256 бит.

Симметричные шифры обладает рядом недостатков:

- Ключ должен храниться в тайне обеими сторонами и передаваться только по *защищенному* каналу (либо в зашифрованном виде). Это требует дополнительных ресурсов.
- Шифровать могут только те, кто знает ключ. Это означает либо жесткие ограничения на количество шифрующих лиц, либо значительный риск компрометации ключа в случае ослабления этих ограничений (компрометация ключа – его раскрытие не криптографическим способом [43]).

1.1.4. Асимметричная криптография

После появления понятия односторонней функции появился другой вид алгоритмов – асимметричные. Такой способ шифрования предполагает наличие известного всем участникам схемы шифрования *открытого ключа* и известного только законным участникам *закрытого* ключа. Открытый ключ используется для шифрования данных, а закрытый – для расшифровки. Таким образом, шифровать сообщения может кто угодно, а вот расшифровывать – только законные пользователи.

Такой способ работы снимает основные проблемы симметричного шифрования: хранить в секрете шифрующий ключ нет необходимости, также он может передаваться по открытым каналам в незашифрованном виде и транслироваться на любую аудиторию.

По причине наличия открытого ключа такие алгоритмы также называются алгоритмами с открытым ключом. Классический пример – алгоритм RSA.

В некоторых случаях, например при использовании цифровой подписи, сообщения шифруются закрытым ключом, а расшифровываются открытым.

1.1.5. Криптографические хэш-функции

Другим направлением криптографии, основанным на понятии односторонней функции, являются криптографические (односторонние, однонаправленные) хэш-функции.

Хэш-функция – функция, принимающая строку произвольной (или почти произвольной) длины, и преобразующую ее в строку фиксированной, обычно меньшей, длины. Полученная строка называется отпечатком (дайджестом) входной строки или ее хэш-кодом.

Однонаправленная хэш-функция вычисляется только в одном направлении: легко вычислить значение дайджеста по входной строке, но крайне трудно создать прообраз, дайджест которого соответствует заданной строке [43].

Криптографическая хэш-функция является открытой. Безопасность обеспечивается именно однонаправленностью функции. Одним из необходимых условий является изменение значения половины битов дайджеста при изменении в одном бите входа: невозможно путем сравнения близких входов обратить хэширующую функцию.

Хэш-функции применяются при вычислении контрольных сумм для проверки подлинности файлов и транзакций: почти невозможно подобрать поддельный файл, отличный от настоящего, но с таким же дайджестом. При таком применении от функции требуется высокое быстродействие, т. к. файлы могут иметь значительный размер, а транзакции часто должны обрабатываться в реальном времени.

Еще одно использование контрольных сумм – при использовании цифровой подписи можно подписывать не весь документ целиком, а только его контрольную сумму. Это значительно ускоряет работу с подписью и уменьшает потребление памяти при хранении подписей [43].

Также криптографические хэш-функции используются для хранения паролей. Такие хэш-функции называются Key Derivation Functions, KDF. В этом случае на носителе пароль не сохраняется, а хранится только его хэш, и каждый раз при вводе строки вычисляется ее хэш-код и сравнивается с хэшем пароля. Если произошло совпадение, значит, считаем, пароль введен верно. Для минимизации вероятности коллизии можно хранить два дайджеста от разных хэш-функций.

Такой способ хранения паролей является в настоящее время наиболее распространенным, так как сам по себе обеспечивает дополнительных уровень защищенности: даже если злоумышленник получил доступ к хэшам, для восстановления по ним паролей ему потребуется много времени (или большие вычислительные мощности). При этом от хэширующей пароли функции требуется, чтобы она вычислялась долго: время вычисления порядка 100 мс незаметно при авторизации, однако значительно затрудняет подбор пароля методом грубой силы. Кроме того, по тем же соображениям желательно большое потребление оперативной памяти, это также усложняет перебор, особенно в многопоточном режиме. К их стойкости также предъявляются повышенные требования.

Для дополнительного повышения защищенности хэшированных паролей используется *соль*. Это случайная строка, добавляемая к шифруемым данным, она должна храниться вместе с хэшем. Она не позволит понять, что захэшированы одинаковые строки, так как они будут иметь разную соль. Благодаря этому, брутфорс одного пароля из базы не позволяет найти другие такие же пароли в базе [43].

Итак, есть два вида криптографических хэш-функций:

- Быстрые, используются для вычисления контрольных сумм и должны потреблять как можно меньше ресурсов. Примеры: MD5, семейство SHA2, из новых – SHA3 и BLAKE2.
- Медленные, используются для хэширования паролей. Должны потреблять много ресурсов и быть очень стойкими. Примеры: bcrypt и scrypt, новая – Argon2.

1.2. Интернет вещей

1.2.1. Основные понятия

Сэмюэл Грингард (Samuel Greengard) – журналист, специализируется на новых технологиях. Директор по маркетингу во многих технологических и бизнес–изданиях, бывший президент Американского общества журналистов. В области IoT известен главным образом книгой [6]. Далее в этом разделе приводятся избранные положения этой работы.

Подключаемые устройства – устройства, которые обмениваются данными по обычному интернет-соединению и получают дополнительные преимущества при подключении, например, через закрытую или частную сеть. Подключаемые устройства необязательно подсоединяются именно к Интернету вещей, но это происходит все чаще.

Радиочастотная идентификация (RFID) – это основной инструмент, который позволяет устройствам стать подключаемыми. Эта технология автоматической идентификации основана на считывании или записи данных, хранящихся в RFID–метках. RFID-метки могут быть как активными (с собственным источником питания), так и пассивными (им не требуется источник питания). И те, и другие позволяют считывателям автоматически получать сигнал и данные с меток. При этом метка должна находиться не дальше определенного допустимого расстояния.

Пассивные радиочастотные метки особенно востребованы благодаря низкой стоимости, долговечности и отсутствию необходимости в постоянном электропитании, они получают питание от ближайшего считывателя. Они могут быть встроены в наклейки для удобства использования или имплантированы под кожу (например, продукт VeriChip [50]). Стоимость определяется мощностью считывающего устройства. Чем больше его мощность, тем меньше требования к размеру и качеству метки, что, в свою очередь, определяет более низкую стоимость [3]. Уже в 2004 году некоторые метки стоили всего 5 центов [48].

Промышленный Интернет – оборудование и аппаратура, оборудованные датчиками. Датчики позволяют получать данные с самых различных устройств унифицированным образом. Затем данные со считывающих устройств могут быть собраны и централизованно обработаны.

Таким образом, возможность сделать большинство устройств подключаемыми позволяет централизованно (в смысле единого канала, Интернета) управлять

практически всей техникой. В результате возникает *Интернет всего* (этот термин введен компанией Cisco). Интернет всего позволяет значительно увеличить автоматизацию за счет упрощения взаимодействия отдельных устройств, модулей и систем. В конечном счете это повышает управляемость системой и, если необходимо, целым кластером систем.

В книге упоминается «парадокс автоматизации»: по мере развития автоматизированных систем вероятность аварии или сбоя снижается, однако степень тяжести потенциальной опасности во много раз повышается. Это создает дополнительные требования к персоналу и пользователям таких систем. Кроме того, возникает психологический эффект расслабления, когда человек целиком полагается на технику.

Автор отмечает дилемму, стоящую перед разработчиками. Создание функциональных интерфейсов и средств управления делают устройства (и, если посмотреть шире, системы) удобнее, но также делают их мишенью для атак. И если прямого доступа к управлению отдельными модулями нет (или нет соответствующих навыков у пользователя или обслуживающего персонала), неисправность или уязвимость не могут быть обнаружены до тех пор, пока проблема не проявится сама, причинив немалый ущерб. Следовательно, разработчики и производители должны находить новые способы обеспечения безопасности как системы в целом, так и отдельных модулей. При этом система защиты должна иметь максимально простой интерфейс управления или, еще лучше, вовсе не требовать управления.

1.2.2. Уровневая модель IoT и проблемы безопасности

Стоит отметить хорошую работу [34], посвященную обзору элементов сети IoT, ее «слоистой» архитектуры, а также актуальных проблем безопасности. Этот раздел написан в основном по материалам данной работы.

Для идентификации устройств используются адреса IPv4 и IPv6, для именования может использоваться система ucode. Сбор информации осуществляется, например, с помощью RFID-меток, носимых устройства. Для сетевого взаимодействия – основного компонента IoT – используются такие технологии, как RFID, NFC, Bluetooth, Wi-Fi, LTE. Обработка информации включает в себя отбор информации и затем необходимые вычисления. Примеры аппаратных платформ: Arduino, Raspberry Pi, Intel Galileo. Среди ОС можно назвать TinyOS, LiteOS,

Android. Для обеспечения общей семантики используются унифицированные модели представления данных RDF, OWL, EXI.

Приложения IoT предоставляют следующие 4 типа услуг пользователю:

– распределенное распознавание объектов;
– сбор, обработка и распределенное хранение информации;
– распределенная решающая (управляющая) система;
– контроль и динамическая корректировка работы различных устройств.

Архитектура сети IoT состоит из нескольких уровней. Базовый трехуровневый вариант отражает главные черты архитектуры. Он состоит из следующих уровней:

– прикладной уровень (application layer);
– сетевой уровень (network layer, transmission layer);
– сенсорный уровень (perception layer, sensor layer).

Сенсорный уровень. На этом уровне осуществляется идентификация «вещей» и сбор поставляемых ими данных. Данные могут быть самыми разными, в зависимости от конкретной области применения: местоположение, температура, вибрация и другие. Этот уровень зачастую является основной целью злоумышленников. См, например, статью [29]. Основные атаки:

– Подслушивание (eavesdropping) – перехват информации. Для таких атак уязвимы данные, передаваемые в открытом или слабо зашифрованном виде.
– Захват узла (node capture) – захват контроля над важным узлом передачи или временного хранения данных. Может привести к утечке сразу большого количества информации, в том числе ключей шифрования.
– Введение фальшивого узла (fake and malicious node) – добавление нового узла в систему. Эта атака направлена на прекращение передачи реальной информации и передачу вместо нее фальшивой. Кроме того, узел может пытаться получить контроль над другими узлами или заставить их повысить энергопотребление для причинения максимального урона сети.
– Атака повторного воспроизведения (атака повторением пакетов, replay attack, playback attack) – злоумышленник собирает информацию, передаваемую от отправителя к получателю, и фиксирует реакцию получателя. После чего позднее отправляет такую же информацию с целью заставить получателя выполнить желаемые действия. Эта атака весьма

– Атака по времени (timing attack) – злоумышленники фиксирует время реакции системы на различные запросы, чтобы получить сведения о ее устройстве и обнаружить потенциальные уязвимости. Может быть также направлена на одно устройство, в этом случае наиболее удобны слабые устройства (для них различие во времени зафиксировать легче).

Сетевой уровень. Он выступает как промежуточное звено между сенсорным уровнем и прикладным уровнем. Занимается передачей информации, а также отвечает за взаимодействие различных устройств и сетей между собой. Он уязвим для следующих атак:

– DoS-атака – создание условий для прекращения или затруднения доступа пользователей к вычислительной сети. Обычно это достигается путем искусственного наводнения сети большим количеством запросов.
– Атака посредника (man in the middle, MITM) – ретрансляция и изменение связи между узлами, которые считают, что общаются друг с другом. Является серьезной угрозой, так как атака может быть осуществлена в реальном времени.
– Атака на хранилище (storage attack) – как при централизованном, так и при распределенном способе хранения информации, она может быть украдена, подделана или удалена.
– Эксплойт-атака (exploit attack) – внедрение фрагментов кода, использующих уязвимости в системе безопасности, в приложении или в аппаратном обеспечении. Целью атаки является получение над контролем системы и кража информации, также возможно нарушение функционирования системы.

На прикладном уровне находятся все приложения, использующие технологию IoT. Атаки на данный уровень часто определяются конкретным назначением сети. Примеры атак:

– Межсайтовый скриптинг (cross-site scripting) – введение инъекции на стороне клиента. Эта атака позволяет злоумышленнику полностью изменить содержимое приложения в своих целях, а также украсть информацию.
– Атака вредоносного кода (malicious code attack) – вредоносный код в любой части приложения. Данные атаки часто предотвращаются антивирусом.
– Атаки на данные – в случае большого количества пользователей, в некоторых случаях данные могут передаваться в слабо защищенном виде, что может быть использовано злоумышленником.

С развитием технологии IoT, распространение получает более сложная, пятиуровневая модель. Модель выглядит следующим образом.

- уровень бизнес-логики (business level);
- прикладной уровень (application layer);
- уровень обработки (processing layer);
- транспортный уровень (transport layer);
- сенсорный уровень (perception layer, sensor layer).

Уровень обработки собирает, отбирает и обрабатывает информацию, полученную от транспортного уровня. Возможные атаки:

- Атака истощения ресурсов (resource exhaustion attack) – атака, в результате которой происходит не «зависание» и перегрузка устройств, как при DoS-атаке, а сбои программного или аппаратного обеспечения системы. Благодаря распределенной природе системы интернета вещей, эти атаки не слишком опасны.
- Вредоносное ПО – атака на конфиденциальную информацию. Вирусы, шпионское ПО, реклама, троянские программы и черви.

Уровень бизнес-логики занимается управлением всей системой. Устанавливает политики приложений, конфиденциальности данных, обработки данных. Уязвимость этого слоя позволит злоумышленникам «легально» использовать приложения в обход бизнес-логики. Возможные варианты атак:

- Атака на бизнес-логику (business logic attack) – использование ошибок программирования бизнес-уровня. Позволяет изменить взаимодействие между пользователем и БД приложения в сторону, выгодную злоумышленнику. Это может быть достигнуто при использовании уязвимостей кода, слабостей процедур валидации при восстановлении пароля и введении входных данных, или слабостей шифрования.
- Атака нулевого дня (zero-day attack) – использование уязвимостей системы безопасности, которые ранее были неизвестны.

1.3. Выводы

В данной главе была приведена основная терминологеская и теоретическая база, на которой будут строиться дальнейшие главы. Рассмотрены основные понятия криптографии и интернета вещей. Описана уровневая модель систем интернета вещей, рассмотрен каждый уровень и атаки на него.

ГЛАВА 2. НАЗВАНИЕ ВТОРОЙ ГЛАВЫ: РАЗРАБОТКА МЕТОДА, АЛГОРИТМА, МОДЕЛИ ИССЛЕДОВАНИЯ

Глава посвящена более подробным примерам оформления текстово-графических объектов.

В параграфе 2.1 приведены примеры оформления многострочной формулы и одиночного рисунка. Параграф 2.2 раскрывает правила оформления перечислений и псевдокода. В параграфе 2.3 приведены примеры оформления сложносоставных рисунков, длинных таблиц, а также теоремоподобных окружений.

2.1. Название параграфа

Все формулы, размещенные в отдельных строках, подлежат нумерации, например, как формулы (2.1) и (2.2) из [28].

$$A^{\uparrow} = \{m \in M \mid gIm \; \forall g \in A\}; \tag{2.1}$$

$$B^{\downarrow} = \{g \in G \mid gIm \; \forall m \in B\}. \tag{2.2}$$

Обратим внимание, что формулы содержат знаки препинания и что они выровнены по левому краю (с помощью знака `&` окружения `align`).

На рис.2.1 приведёна фотография Нового научно-исследовательского корпуса СПбПУ.

Рис.2.1. Новый научно-исследовательский корпус СПбПУ [45]

2.2. Название параграфа

Название параграфа оформляется с помощью команды `\section{...}`, название главы — `\chapter{...}`.

2.2.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`.

Использование подподпараграфов в основной части крайне не рекомендуется. В случае использования, необходимо вынести данный номер в содержание. Название подпараграфа оформляется с помощью команды `\subsubsection{...}`.

Вместо подподпараграфов рекомендовано использовать перечисления.

Перечисления могут быть с нумерационной частью и без неё и использоваться с иерархией и без иерархии. Нумерационная часть при этом формируется следующим способом:

1. в перечислениях *без иерархии* оформляется арабскими цифрами с точкой (или длинным тире).
2. В перечислениях *с иерархией* — в последовательности сначала прописных латинских букв с точкой, затем арабских цифр с точкой и далее — строчных латинских букв со скобкой.

Далее приведён пример перечислений с иерархией.

A. Первый пункт.
B. Второй пункт.
C. Третий пункт.
D. По ГОСТ 2.105–95 [7] первый уровень нумерации идёт буквами русского или латинского алфавитов (*для определенности выбираем английский алфавит*), а второй — цифрами.
    1. В данном пункте лежит следующий нумерованный список:
        a) первый пункт;
        b) третий уровень нумерации не нормирован ГОСТ 2.105–95 (*для определенности выбираем английский алфавит*);
        c) обращаем внимание на строчность букв в этом нумерованном и следующем маркированном списке:
            – первый пункт маркированного списка.
E. Пятый пункт верхнего уровня перечисления.

Маркированный список (без нумерационной части) используется, если нет необходимости ссылки на определенное положение в списке:

– первый пункт с *маленькой буквы* по правилам русского языка;
– второй пункт с *маленькой буквы* по правилам русского языка.

Оформление псевдокода необходимо осуществлять с помощью пакета `algorithm2e` в окружении `algorithm`. Данное окружение интерпретируется в шаблоне как рисунок. Пример оформления псевдокода алгоритма приведён на рис.2.2.

Algorithm

Input: the many-valued context $\mathbb{M} \stackrel{\text{def}}{=} (G,M,W,J)$, the class membership $\varepsilon : G \to K$

Output: positive and negative binary contexts $\overline{\mathbb{K}_+} \stackrel{\text{def}}{=} (\overline{G_+},M,I_+)$, $\overline{\mathbb{K}_-} \stackrel{\text{def}}{=} (\overline{G_-},M,I_-)$ such that i-tests found in $\overline{\mathbb{K}_+}$ are diagnostic tests in $\mathbb{M}$, and objects from $\overline{\mathbb{K}_-}$ are counter-examples

1. **for** $\forall g_i, g_j \in G$ **do**
2. **if** $i < j$ **then**
3. $\overline{G} \leftarrow (g_i,g_j)$;
4. **for** $\forall (g_i,g_j) \in \overline{G}$ **do**
5. **if** $m(g_i) = m(g_j)$ **then**
6. $(g_i,g_j)Im$;
7. **if** $\varepsilon(g_i) = \varepsilon(g_j)$ **then**
8. $\overline{G_+} \leftarrow (g_i,g_j)$;
9. **else** $\overline{G_-} \leftarrow (g_i,g_j)$;
10. $I_+ = I \cap (\overline{G_+} \times M)$, $I_- = I \cap (\overline{G_-} \times M)$;
11. **for** $\forall \overline{g_+} \in \overline{G_+}$, $\forall \overline{g_-} \in \overline{G_-}$ **do**
12. **if** $\overline{g_+}^{\uparrow} \subseteq \overline{g_-}^{\uparrow}$ **then**
13. $\overline{G_+} \leftarrow \overline{G_+} \setminus \overline{g_+}$;

Рис.2.2. Псевдокод алгоритма `DiagnosticTestsScalingAndInferring` [41]

Обратим внимание, что можно сослаться на строчку 1 псевдокода из рис.2.2.

2.3. Название параграфа

Одиночные формулы также, как и отдельные формулы в составе группы, могут быть размещены в несколько строк. Чтобы выставить номер формулы напротив средней строки, используйте окружение `multlined` из пакета `mathtools`

следующим образом [28]:

$$\begin{aligned}(A_1,B_1) \leqslant (A_2,B_2) \;\; &\Leftrightarrow \\ \Leftrightarrow \;\; A_1 \subseteq A_2 \;\; &\Leftrightarrow \\ \Leftrightarrow \;\; B_2 \subseteq B_1&.\end{aligned} \tag{2.3}$$

Используя команду `\labelcref{...}` из пакета `cleveref`, допустимо оформить ссылку на несколько формул, например, (2.1–2.3).

Пример оформления четырёх иллюстраций в одном текстово-графическом объекте приведён на рис.2.3. Это возможно благодаря использованию пакета `subcaption`.

a)

b)

c)

d)

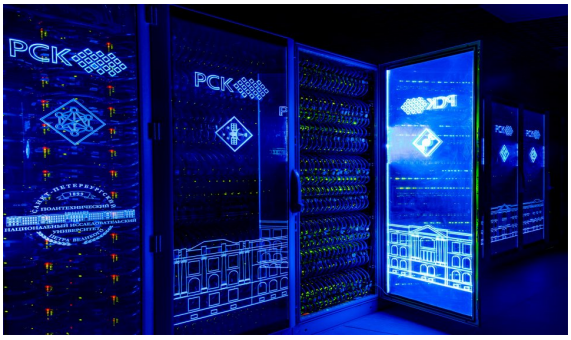

Рис.2.3. Фотографии суперкомпьютерного центра СПбПУ [45]: *a* — система хранения данных и узлы NUMA-вычислителя; *b* — холодильные машины на крыше научно-исследовательского корпуса; *c* — машинный зал; *d* — элементы вычислительных устройств

Далее можно ссылаться на составные части данного рисунка как на самостоятельные объекты: рис.2.3a, рис.2.3b, рис.2.3c, рис.2.3d или на три из четырёх изображений одновременно: рис.2.3a–2.3c.

Приведём пример табличного представления данных с записью продолжения на следующей странице на табл.2.1.

Таблица 2.1

Пример задания данных из [42] (с повтором для переноса таблицы на новую страницу)

G	m_1	m_2	m_3	m_4	K
1	2	3	4	5	6
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1

Продолжение табл. 2.1

1	2	3	4	5	6
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.2

Пример представления данных для сквозного примера по ВКР [42]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.3

Пример задания данных в табличном виде из [42] (с помощью окружения minipage)

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Рис.2.4. Новый научно-исследовательский корпус СПбПУ [45] (с помощью окружения minipage)

Вопросы форматирования текстово-графических объектов (окружений) не регламентированы в известных нам ГОСТах, поэтому предлагаем придерживаться следующих правил:

- **полужирный текст** рекомендуем использовать только для названий стандартных окружений с нумерационной частью, например, для представления *впервые*: **определение 1.1**, **теорема 2.2**, **пример 2.3**, **лемма 4.5**;
- *курсив* рекомендуем использовать только для выделения переменных в формулах, служебной информации об авторах главы (статьи), важных терминов, представляемых по тексту, а также для всего тела окружений, связанных с получением *новых существенных результатов и их доказательством*: теорема, лемма, следствие, утверждение и другие.

По аналогии с нумерацией формул, рисунков и таблиц нумеруются и иные текстово-графические объекты, то есть включаем в нумерацию номер главы, например: теорема 3.1. для первой теоремы третьей главы монографии. Команды LaTeX выставляют нумерацию и форматирование автоматически. Полный перечень команд для подготовки текстово-графических и иных объектов находится в подробных методических рекомендациях [15].

Для удобства авторов названия стандартных окружений, рекомендованных к использованию, приведены в табл.2.4, а в табл.2.5 перечислены имена специально разработанных окружений для шаблонов SPbPU.

На базе пакета `tikz` разработано большое количество расширений [25], например, `tikzcd`, которые мы рекомендуем использовать для оформления иллюстраций.

В случае, если авторам потребовалось новое окружение, то создать его можно в файле в файле `my_folder/my_settings.tex` согласно правилам, приведённым ниже.

1. Для перехода в режим создания окружений следует указать:
   - `\theoremstyle{myplain}` — окружения с доказательствами или аксиомами
   - `\theoremstyle{mydefinition}` — окружения, не связанные с доказательствами или аксиомами.
2. В команде создания окружения следует ввести краткий псевдоним (`m-new-env`) и отображаемое в pdf имя окружения (`Название_окружения`):
   - `\newtheorem{m-new-env-second}{Название_окружения}[chapter]`.

Таблица 2.4

Стандартные окружения

Название окружения	Назначение
`center`	центрирование, аналог команды `\centering`, но с добавлением нежелательного пробела, поэтому лучше избегать применения `center`
`itemize`	перечисления, в которых нет необходимости нумеровать пункты (немаркированные списки)
`enumerate`	перечисления с нумерацией (немаркированные списки)
`refsection`	создание отдельных библиографических списков для глав
`tabular`	оформление таблиц
`table`	автоматическое перемещение по тексту таблиц, оформленных, например, с помощью `tabular`, для минимизации пустых пространств
`longtable`	оформление многостраничных таблиц
`tikzpicture`	создание иллюстраций с помощью пакета `tikz` [25]
`figure`	автоматическое перемещение по тексту рисунков, оформленных например, с помощью `tikz` или подключенных с помощью команды `\includegraphics`, для минимизации пустых пространств
`subfigure`	оформление вложенных рисунков в составе `figure`
`algorithm`	оформление псевдокода на основе пакета `algorithm2e` [26]
`minipage`	оформление рисунков и таблиц без функций автоматического перемещения по тексту для минимизации пустых пространств
`equation`	оформление выключенных (не встроенных в текст с помощью `$...$`) одиночных формул на одной строке
`multilined`	оформление выключенных (не встроенных в текст с помощью `$...$`) одиночных формул в несколько строк
`aligned`	оформление нескольких формул с выравниванием по символу `&`.

Таблица 2.5

Специальные окружения

Название окружения	Текстово-графический объект
`abstr`	реферат (abstract)
`m-theorem`	теорема
`m-corollary`	следствие
`m-proposition`	утверждение
`m-lemma`	лемма
`m-axiom`	аксиома
`m-example`	пример
`m-definition`	определение
`m-condition`	условие
`m-problem`	проблема
`m-exercise`	упраженение
`m-question`	вопрос
`m-hypothesis`	гипотеза

Теорема 2.1 (о чем-то конкретном)**.** *Текст теоремы полностью выделен курсивом. Допустимо математические символы не выделять курсивом, если это искажает их значения. Используется абзацный отсуп, так как "Абзацы в тексте начинают отступом" в соответствии с ГОСТ 2.105–95. Название теоремы допустимо убрать. Доказательство окончено.*

Доказательство теоремы 2.1, леммы, утверждений, следствий и других подобных окружений (в последнем абзаце) завершаем предложением в котором сказано, что доказательство окончено. Например, доказательство теоремы 2.1 окончено.

Тело доказательства не выделяется курсивом. Тело следующих окружений также не выделяется сплошным курсивом: определение, условие, проблема, пример, упражнение, вопрос, гипотеза и другие.

Определение 2.1 (термин)**.** В тексте определения только *важные термины* выделяются курсивом. Если определение носит лишь вспомогательный характер, то допустимо не использовать окружение `m-definition`, представляя текст определения в обычном абзаце. Ключевые термины при этом обязательно выделяются курсивом.

Вместо теоремо-подобных окружений для вставки небольших текстово-графических объектов иногда используются команды. Типичным примером такого подхода является команда `\footnote{text}`[4], где в аргументе `text` указывают текст *подстрочной ссылки (сноски)*.В них *нельзя добавлять веб-ссылки или цитировать литературу*. Для этих целей используется список литературы. Нумерация сносок сквозная по ВКР без точки на конце выставляется в шаблоне автоматически, однако в каждом приложении к ВКР нумерация, зависящая от номера приложения, выставляется префикс «П», например «П1.1» — первая сноска первого приложения.

2.4. Выводы

Текст заключения ко второй главе. Пример ссылок [**Article**; 16; 17; 19; 22—24; 27; 31; 35; 36; 38; 39; 49], а также ссылок с указанием страниц, на котором отображены те или иные текстово-графические объекты [41, с. 96] или в виде мультицитаты на несколько источников [41, с. 96; 28, с. 46]. Часть библиографиче-

[4]Внимание! Команда вставляется непосредственно после слова, куда вставляется сноска (без пробела). Лишние пробелы также не указываются внутри команды перед и после фигурных скобок.

ских записей носит иллюстративный характер и не имеет отношения к реальной литературе.

Короткое имя каждого библиографического источника содержится в специальном файле `my_biblio.bib`, расположенном в папке `my_folder`. Там же находятся исходные данные, которые с помощью программы `Biber` и стилевого файла `Biblatex-GOST` [20] приведены в списке использованных источников согласно ГОСТ 7.0.5-2008. Многообразные реальные примеры исходных библиографических данных можно посмотреть по ссылке [21].

Как правило, ВКР должна состоять из четырех глав. Оставшиеся главы можно создать по образцу первых двух и подключить с помощью команды `\input` к исходному коду ВКР. Далее в приложении 1 приведены краткие инструкции запуска исходного кода ВКР [40; 47].

В приложении 2 приведено подключение некоторых текстово-графических объектов. Они оформляются по приведенным ранее правилам. В качестве номера структурного элемента вместо номера главы используется «П» с номером главы. Текстово-графические объекты из приложений не учитываются в реферате.

ГЛАВА 3. НАЗВАНИЕ ТРЕТЬЕЙ ГЛАВЫ: РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

3.1. Название параграфа

3.2. Название параграфа

3.3. Выводы

Текст выводов по главе 3.

ГЛАВА 4. НАЗВАНИЕ ЧЕТВЁРТОЙ ГЛАВЫ. АПРОБАЦИЯ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ, А ИМЕННО: МЕТОДА, АЛГОРИТМА, МОДЕЛИ ИССЛЕДОВАНИЯ

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

4.1. Название параграфа

4.2. Название параграфа

Пример ссылки на литературу [1; 11; 13; 37].

4.3. Выводы

Текст выводов по главе 4.

ЗАКЛЮЧЕНИЕ

Заключение (2 – 5 страниц) обязательно содержит выводы по теме работы, *конкретные предложения и рекомендации* по исследуемым вопросам. Количество общих выводов должно вытекать из количества задач, сформулированных во введении выпускной квалификационной работы.

Предложения и рекомендации должны быть органически увязаны с выводами и направлены на улучшение функционирования исследуемого объекта. При разработке предложений и рекомендаций обращается внимание на их обоснованность, реальность и практическую приемлемость.

Заключение не должно содержать новой информации, положений, выводов и т. д., которые до этого не рассматривались в выпускной квалификационной работе. Рекомендуется писать заключение в виде тезисов.

Последним абзацем в заключении можно выразить благодарность всем людям, которые помогали автору в написании ВКР.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

DOI Digital Object Identifier.

WoS Web of Science.

ВКР Выпускная квалификационная работа.

ТГ-объект Текстово-графический объект.

СЛОВАРЬ ТЕРМИНОВ

TeX — язык вёрстки текста и издательская система, разработанные Дональдом Кнутом.

LaTeX — язык вёрстки текста и издательская система, разработанные Лэсли Лампортом как надстройка над TeX.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Автономова Н. С.* Философский язык Жака Деррида. — М.: Российская политическая энциклопедия (РОССПЭН), 2011. — 510 с. — (Сер.: Российские Пропилеи).

2. *Беклемищева Н.*, *Жуков В.*, *Михайлов К.* О некоторых вопросах информационной безопасности медицинских устройств // Актуальные проблемы авиации и космонавтики. — 2017. — № 13.

3. *Бхуптани М.*, *Морадпур Ш.* RFID-технологии на службе вашего бизнеса / пер. Н. Троицкий. — Москва: Альпина Паблишер, 2007. — 290 с.

4. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. — URL: http://docs.cntd.ru/document/gost-r-50922-2006 (дата обращения: 10.06.2020).

5. ГОСТ Р 53114-2008. — URL: http://docs.cntd.ru/document/gost-r-53114-2008 (дата обращения: 10.06.2020).

6. *Грингард С.* Интернет вещей: Будущее уже здесь / пер. М. Трощенко. — Москва: Альпина Паблишер, 2017. — 188 с.

7. Единая система конструкторской документации. Общие требования к текстовым документам [текст]: ГОСТ 2.105–95. — Взамен ГОСТ 2.105—79, ГОСТ 2.906—71 ; введ. 1996—07—01. — Минск : Межгос. совет по стандартизации, метрологии и сертификации, 2002. — 31 с. — (Сер.: Межгосударственный стандарт).

8. Защита для чайников. Интернет вещей хотят обезопасить от кибератак. — URL: https://rg.ru/2016/10/13/evrokomissiia-predlozhila-zashchitit-internet-veshchej-ot-kiberatak.html (дата обращения: 10.06.2020).

9. Интернет вещдоков. Власти разработали концепцию развития сетей IoT. — URL: https://www.kommersant.ru/doc/3924324 (дата обращения: 10.06.2020).

10. Информация – Большая советская энциклопедия. — URL: https://gufo.me/dict/bse/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F (дата обращения: 10.06.2020).

11. *Котельников И. А.*, *Чеботаев П. З.* LaTeX по-русски. — 3-е изд. — Новосибирск: Сибирский Хронограф, 2004. — 496 с. — URL: http://www.tex.uniyar.ac.ru/doc/kotelnikovchebotaev2004b.pdf (дата обращения: 06.03.2019).

12. *Музыкантский А.*, *Фурин В.* Лекции по криптографии. — 2-е изд. — Москва: МЦНМО, 2013. — 68 с.

13. *Песков Н. В.* Поиск информативных фрагментов описаний объектов в задачах распознавания: дис. . . . канд. канд. физ.-мат. наук: 05.13.17 / Песков Николай Владимирович. — М., 2004. — 102 с.

14. *Ященко В. В.* Введение в криптографию. — 4-е изд. — Москва: МЦНМО, 2012. — 348 с.

15. Author and editor guide to prepare and submit the academic SPbPU editions to Clarivate Analytics: Book Citation Index Web of Science / V. Parkhomenko [et al.]. — 2018. — URL: https://github.com/ParkhomenkoV/SPbPU-BCI-template/blob/master/Author_guide_SPbPU-BCI.pdf (visited on 06.03.2019).

16. *Babington P.* The title of the work. Vol. 4. — 3rd ed. — The address: The name of the publisher, 1993. — 255 p. — (Ser.: 10).

17. *Badiou A.* Briefings on Existence: A Short Treatise on Transitory Ontology / ed. and trans. from the French, with an introd., by N. Madarasz. — NY: SUNY Press, 2006. — 190 p. — URL: https://books.google.ru/books?id=7HNkAT%5C_NFksC (visited on 05.12.2017).

18. *Bauer F. L.* Decrypted Secrets: Methods and Maxims of Cryptology. — 4th ed. — Berlin: Springer, 2007. — 555 p.

19. *Caxton P.* The title of the work. — The address of the publisher, 1993. — 255 p.

20. *Domanov O.* BibLATEX support for GOST standard bibliographies. — URL: https://ctan.org/pkg/biblatex-gost (visited on 06.03.2019).

21. *Domanov O.* Biblatex-GOST examples. — URL: http://ctan.altspu.ru/macros/latex/contrib/biblatex-contrib/biblatex-gost/doc/biblatex-gost-examples.pdf (visited on 06.03.2019).

22. *Draper P.* The title of the work // The title of the book. Vol. 4 / ed. by T. editor. — The organization. The address of the publisher: The publisher, 1993. — (Ser.: 5).

23. *Eston P.* The title of the work // Book title. Vol. 4. — 3rd ed. — The address of the publisher: The name of the publisher, 1993. — Chap. 8 — P. 201–213. — (Ser.: 5).

24. *Farindon P.* The title of the work // The title of the book. Vol. 4 / ed. by T. editor. — 3rd ed. — The address of the publisher: The name of the publisher, 1993. — Chap. 8 — P. 201–213. — (Ser.: 5).

25. *Feuersanger C.*, *Tantau T.* The TikZ and PGF packages. — URL: https://ctan.org/pkg/pgf (visited on 06.03.2019).

26. *Fiorio C.* The algorithm2e package. — URL: https://ctan.org/pkg/algorithm2e (visited on 06.03.2019).

27. *Gainsford P.* The title of the work / The organization. — 3rd ed. — The address of the publisher, 1993. — 255 p.

28. *Ganter B.*, *Wille R.* Formal concept analysis: mathematical foundations. — Springer, Berlin, 1999. — 284 p.

29. *Giannetsos T.*, *Dimitriou T.*, *Prasad N.* Weaponizing Wireless Networks: An Attack Tool for Launching Attacks against Sensor Networks //. — 2010.

30. Government response to the Regulatory proposals for consumer Internet of Things (IoT) security consultation. — URL: https://www.gov.uk/government/consultations/consultation-on-regulatory-proposals-on-consumer-iot-security/outcome/government-response-to-the-regulatory-proposals-for-consumer-internet-of-things-iot-security-consultation (visited on 10.06.2020).

31. *Harwood P.* The title of the work: Master's thesis / Harwood Peter. — The address of the publisher: The school where the thesis was written, 1993. — 255 p.

32. HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack. — URL: https://www8.hp.com/us/en/hp-news/press-release.html?id=1744676 (visited on 10.06.2020).

33. Internet Of Things (iot). — URL: https://www.gartner.com/en/information-technology/glossary/internet-of-things (visited on 10.06.2020).

34. IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey / M. Burhan [et al.] // Sensors. — 2018.

35. *Isley P.* The title of the work. — 1993.

36. *Joslin P.* The title of the work: diss. … PhD in Engineering / Joslin Peter. — The address of the publisher: The school where the thesis was written, 1993. — 255 p.

37. *Kotelnikov I. A.*, *Chebotaev P. Z.* LaTeX in Russian. — 3rd ed. — Novosibirsk: Sibiskiy Hronograph, 2004. — 496 p. — URL: http://www.tex.uniyar.ac.ru/doc/kotelnikovchebotaev2004b.pdf (visited on 06.03.2019); (in Russian).

38. *Lambert P.* The title of the work: tech. rep. / The institution that published. — The address of the publisher, 1993. — 255 p. — No. 2.

39. *Marcheford P.* The title of the work. — 1993.

40. MiKTeX web site. — URL: https://miktex.org/ (visited on 06.03.2019).

41. Notes on relation between symbolic classifiers / X. Naidenova [et al.] // CEUR Workshop Proceedings / ed. by K. S. Watson B.W. — 2017. — Vol. 1921. — P. 88–103. — URL: http://ceur-ws.org/Vol-1921/paper9.pdf (visited on 19.12.2017).

42. *Peskov N. V.* Searching for informative fragments of object descriptions in the recognition tasks: diss. ... cand. phys.-math. sci.: 05.13.17 / Peskov Nickolay Vladimirovich. — M., 2004. — 102 p. — (in Russian).

43. *Schneier B.* Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth). — 1st ed. — John Wiley, Sons, Inc, 1996. — 1027 p.

44. Secrecy (Obscurity) is a Valid Security Layer. — URL: https://danielmiessler.com/study/security-by-obscurity/ (visited on 10.06.2020).

45. SPbPU photo gallery. — URL: http://www.spbstu.ru/media/photo-gallery/ (visited on 06.03.2019).

46. State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. — URL: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ (visited on 10.06.2020).

47. TeXstudio web site. — URL: https://www.texstudio.org/ (visited on 06.03.2019).

48. The 5-Cent RFID Tag. — URL: https://www.rfidjournal.com/the-5-cent-rfid-tag (visited on 10.06.2020).

49. The title of the work. Vol. 4 / ed. by P. Kidwelly. — The organization. The address of the publisher: The name of the publisher, 1993. — 255 p. — (Ser.: 5).

50. VeriChip RFID Implants in Mexican Attorney General's Office Overstated. — URL: https://www.webcitation.org/5wa3VneDo?url=http://www.spychips.com/press-releases/mexican-implant-correction.html (visited on 10.06.2020).

Приложение 1
Краткие инструкции по настройке издательской системы LaTeX

В SPbPU-BCI-template *автоматически выставляются необходимые настройки и в исходном тексте шаблона приведены примеры оформления текстово-графических объектов*, поэтому авторам достаточно заполнить имеющийся шаблон текстом главы (статьи), не вдаваясь в детали оформления, описанные далее. Возможный «быстрый старт» оформления главы (статьи) под Windows следующий[П1.1]:

A. Установка полной версии MikTeX [40]. В процессе установки лучше выставить параметр доустановки пакетов «на лету».
B. Установка TexStudio [47].
C. Запуск TexStudio и компиляция `my_chapter.tex` с помощью команды «Build&View» (например, с помощью двойной зелёной стрелки в верхней панели). *Иногда, для достижения нужного результата необходимо несколько раз скомпилировать документ.*
D. В случае, если не отобразилась библиография, можно
   - воспользоваться командой Tools → Commands → Biber, затем запустив Build&View;
   - настроить автоматическое включение библиографии в настройках Options → Configure TexStudio → Build → Build&View (оставить по умолчанию, если сборка происходит слишком долго): `txs:///pdflatex | txs:///biber | txs:///pdflatex | txs:///pdflatex | txs:///view-pdf`.

В случае возникновения ошибок, попробуйте скомпилировать документ до последних действий или внимательно ознакомьтесь с описанием проблемы в log-файле. Бывает полезным переход (по подсказке TexStudio) в нужную строку в pdf-файле или запрос с текстом ошибке в поисковиках. Наиболее вероятной проблемой при первой компиляции может быть отсутствие какого-либо установленного пакета LaTeX.

В случае корректной работы настройки «установка на лету» все дополнительные пакеты будут скачиваться и устанавливаться в автоматическом режиме. Если доустановка пакетов осуществляется медленно (несколько пакетов за один запуск

[П1.1] Внимание! Пример оформления подстрочной ссылки (сноски).

компилятора), то можно попробовать установить их в ручном режиме следующим образом:

1. Запустите программу: меню → все программы → MikTeX → Maintenance (Admin) → MiKTeX Package Manager (Admin).
2. Пользуясь поиском, убедитесь, что нужный пакет присутствует, но не установлен (если пакет отсутствует воспользуйтесь сначала MiKTeX Update (Admin)).
3. Выделив строку с пакетом (возможно выбрать несколько или вообще все неустановленные пакеты), выполните установку Tools → Install или с помощью контекстного меню.
4. После завершения установки запустите программу MiKTeX Settings (Admin).
5. Обновите базу данных имен файлов Refresh FNDB.

Для проверки текста статьи на русском языке полезно также воспользоваться настройками Options → Configure TexStudio → Language Checking → Default Language. Если русский язык «ru_RU» не будет доступен в меню выбора, то необходимо вначале выполнить Import Dictionary, скачав из интернета любой русскоязычный словарь.

Далее приведены формулы (П1.2), (П1.1), рис.П1.2, рис.П1.1, табл.П1.2, табл.П1.1.

$$\pi \approx 3{,}141. \tag{П1.1}$$

Рис.П1.1. Вид на гидробашню СПбПУ [45]

Таблица П1.1

Представление данных для сквозного примера по ВКР [42]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

П1.1. Параграф приложения

П1.1.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`. Использование подподпараграфов в основной части крайне не рекомендуется.

$$\pi \approx 3{,}141. \tag{П1.2}$$

Рис.П1.2. Вид на гидробашню СПбПУ [45]

Таблица П1.2

Представление данных для сквозного примера по ВКР [42]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Приложение 2

Некоторые дополнительные примеры

В приложении[П2.1] приведены формулы (П2.2), (П2.1), рис.П2.2, рис.П2.1, табл.П2.2, табл.П2.1

$$\pi \approx 3{,}141. \qquad \text{(П2.1)}$$

Рис.П2.1. Вид на гидробашню СПбПУ [45]

Таблица П2.1

Представление данных для сквозного примера по ВКР [42]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

[П2.1] Внимание! Пример оформления подстрочной ссылки (сноски).

П2.1. Подраздел приложения

$$\pi \approx 3{,}141. \qquad \text{(П2.2)}$$

Рис.П2.2. Вид на гидробашню СПбПУ [45]

Таблица П2.2

Представление данных для сквозного примера по ВКР [42]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2