Федеральное агентство связи ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»

Кафедра «‎Информатика»

Курсовая работа по дисциплине «‎Основы программирования»

На тему «Протоколы аутентификации с нулевым разглашением секрета»

Выполнил студент

Группы БСТ2001

Любимов И.С.

Проверил доц. кафедры

«‎Информатика» Гуриков С.Р.

Москва 2020

Содержание что это за разделы такие?? Никто пока такого не делал

[ВВЕДЕНИЕ 3](#_Toc54904160)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc54904161)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 7](#_Toc54904162)

[Раздел 1. Основные понятия 7](#_Toc54904163)

[Раздел 2. Аутентификация пользователей в информационных системах. 14](#_Toc54904168)

[Раздел 3. Протоколы с нулевым разглашением секрета. 18](#_Toc54904171)

[Раздел 4. Двухшаговые протоколы с нулевым разглашением секрета. 20](#_Toc54904174)

[Раздел 5. Протокол на основе схемы открытого согласования ключа 25](#_Toc54904179)

[Раздел 6. Протокол на основе последовательного возведения в квадрат 27](#_Toc54904182)

[Раздел 7. Преобразование протоколов с нулевым разглашением в схемы цифровой подписи 29](#_Toc54904185)

[Раздел 8. Снижение сложности вычислений в криптосхемах, основанных на задаче дискретного логарифмирования. 33](#_Toc54904190)

[Вывод 35](#_Toc54904195)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc54904196)

# ВВЕДЕНИЕ

**Актуальность темы.** В современном мире информация, несомненно, является главным ресурсом. С развитием информационных технологий большинство наших конфиденциальных данных теперь находятся в сети интернет. На данный момент у каждого пользователя есть хотя бы один, а иногда и больше, аккаунтов на различных WEB-серверах. И каждый пользователь хочет быть уверенным, что его **конфиденциальная** информация будет под надежной защитой. Так, программисты поставили перед собой задачу разработки методов защиты. Но с развитием технологий появлялись и новые методы получения конфиденциальной информации в обход **аутентификационных** механизмов. **Аутентификационными** ЗАЧЕМ ЭТО ВСЕ ЖИРНЫМ ВЫДЕЛЯТЬ??

механизмами называют принципы проверки информации о пользователе. Существует множество методов сохранения информации, в частности ее конфиденциальности. **Аутентификационные** механизмы наиболее часто подвергаются атакам злоумышленников. Злоумышленники ищут различные каналы утечки информации, чтобы получить необходимые данные без официального доступа к ним. На данный момент существует несколько протоколов аутентификации:

-протокол с нулевым разглашением секрета,

-двухшаговые протоколы с нулевым разглашением секрета,

-протокол на основе схемы открытого согласования ключа,

-протокол на основе последовательного возведения в квадрат,

-схемы цифровой подписи.

Я НЕ ВИЖУ, ЧТО ИСПОЛЬЗОВАЛСЯ ОБРАЗЕЦ ВВЕДЕНИЯ ГОВОРИЛИ НА ЧТО ОБРАЩАТЬ ВНИМАНИЕ

**Объектом исследования**, приведенным в рамках данной курсовой, являются протоколы аутентификации с нулевым разглашением секрета.

**Предметом исследования** является сам предмет, методы, а также функции протоколов аутентификации с нулевым разглашением секрета.

**Цели работы и задачи исследования.** Целью работы является исследование различных методов аутентификации с нулевым разглашением секрета и на основе выводов разработка алгоритма надежной аутентификации, удобный для практического использования пользователем. Поставленная цель определила следующие задачи исследования:

задачи исследования:

1. Анализ основных теоретических положений.

2. Разработать программный проект, содержащий электронное пособие и текстовую программу по вышеуказанной теме**.**

**Методы исследования.** Для решения поставленных целей и задач были использованы теоретические методы исследования. Теорию исследования составили труды Молдовян А.А., Молдовян Д.Н., Левина А.Б. в области разработки протоколов аутентификационных механизмов с нулевым разглашением секрета.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

**ЗДЕСЬ И ДАЛЕЕ НАДО ЧИТАТЬ ГОСТ В ЧАСТИ ОФОРМЛЕНИЯ ЗАГОЛОВКОВ**

* 1. Основания для разработки

Основанием для разработки является задание, выполненное в соответствии с заданием, полученным от кафедры «Информатика» Московского технического университета связи и информатики и утвержденное научным руководителем доц. Кафедры «Информатика» к.п.н. Гуриковым С.Р. Дата утверждения 02.10.2020.

* 1. Назначение разработки

Программный продукт предназначен для ознакомления с основными теоретическими положениями по теме курсовой работы и проверки знаний пользователя с помощью тестовой программы.

* 1. Требования к программе
     1. Требования к функциональным характеристикам

Разработанный программный продукт должен обеспечить выполнение следующих функций:

* Возможность вывода результатов исследования для пользователя
* Ознакомление с теоретической частью
  + 1. Требования к надежности

Разрабатываемое программное обеспечение должно иметь устойчивую работу, в соответствии с алгоритмом программы, выдавать сообщение об ошибках, поддерживать диалоговый режим, в рамках представляемых пользователю возможностей.

* + 1. Требование к составу и параметрам технических средств

Минимальные и рекомендуемые системные требования для ПК.

* + 1. Требования к информационной и программной совместимости

Программа должна легко устанавливаться, функционировать и корректно работать при наличии следующего ПО: OC Windows XP и более поздние версии.

* + 1. Требования к транспортированию и хранению

Программа поставляется на флешке. Программная документация поставляется в электронном и печатном виде.

* + 1. Требования к программной документации

В ходе разработки программы должны быть подготовлены: текст программы, описание программы, методика испытаний, руководство пользователя.

* 1. Стадии и этапы разработки

Таблица 1 – стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № Этапа | Название этапа | Срок | Отчетность |
|  |  |  |  |

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Раздел 1. Основные понятия

### 1.1 Терминология

* + 1. **Отправитель и получатель**

Одним из основных понятий в **криптографии** являются понятия отправитель и получатель. Допустим, что рядовой отправитель собирается отправить ценное сообщение получателю и, более того, собирается сделать это безопасно: он хочет быть уверен, что перехвативший это сообщение не сможет его прочесть. Эти стороны, которые обмениваются информацией, в **криптографии** называют A и B. Также довольно часто эти аббревиатуры заменяют более простыми для восприятия именами: Алиса и Боб. Но стоит понимать, что участвующие стороны не обязательно должны быть людьми. Используя эти имена, мы вполне можем описывать обмен секретной информацией между двумя автономными механизмами. Имена есть не только у взаимодействующих сторон, например подслушивающей стороне дают имя «Ева» («плохая девочка») – сторона, которая взламывает шифртекст.

#### 1.1.2 Сообщения и шифрование

Само сообщение называется **открытым текстом**. Алгоритм шифрования (или шифр) — это перевод обычного, доступного для понимания текста на зашифрованный (или шифртекст, шифрограмму, криптограмму) с помощью уникального секретного ключа. Данный процесс называют шифрованием. Обозначим открытый текст как M (от message, сообщение). Это может быть что угодно: цифровое видеоизображение, поток битов, растровое(битовое) изображение и т. д. Для компьютера M это неважно —это просто двоичные данные. Какая бы роль не вкладывалась в сообщение в любом случае, M — это сообщение, которое должно быть зашифровано. Обозначим шифртекст как C, это тоже двоичные данные, иногда того же размера, что и M, иногда больше, однако шифрование может сопровождаться сжатием передаваемого сообщения. Но само шифрование не проводит сжатие информации. Функция шифрования E действует на M, создавая C.

Мы будем писать

(1)

где M —открытый текст, Е — шифрующая функция, K —секретный ключи С — шифртекст.

Обратный процесс называют **расшифрованием** и пишут

, (2)

Где M — открытый текст, D — дешифрующая функция, K — секретный ключи С — шифртекст.

Заметим, что алгоритмы шифрования и расшифрования Е и D открыты, и вся секретность, необходимая для шифрования информации зависит от секретного ключа K. Так как основой задачей шифрования и дешифрования является безопасная отправка, шифрование и восстановление первоначального сообщения, тогда будет выполняться следующее неравенство:

, (3)

ТАК, ДАВАЙ ЧИТАЙ ГОСТ

МЫ ДВА МЕСЯЦА ОБ ЭТОМ ГОВОРИМ

КОГДА ВЫУЧИШЬ ЕГО, МОЖЕШЬ КО МНЕ ПОДОЙТИ И РАССКАЗАТЬ

ПРИСЛАННОЕ НЕВОЗМОЖНО СМОТРЕТЬ

Где M — открытый текст, D — дешифрующая функция, K — секретный ключ, Е — шифрующая функция.

Искусство и наука, занимающаяся шифрованием и дешифрованием, называется криптографией. Криптоаналитиками называются те, кто используют криптоанализ, искусство и науку взламывать шифртекст, то есть, раскрывать, то, что было зашифровано. Отрасль математики, которая занимается криптографией и **криптоанализом**, называется **криптологией**, а люди, которые ей занимаются – **криптологами**.



Рисунок 1.1 – Шифрование и дешифрование

#### 1.1.3 Проверка подлинности, целостность и неотрицание авторства

Помимо того, что **криптография** обеспечивает конфиденциальность информации, она также имеет и другие функции:

—Проверка подлинности. В таком случае получатель будет точно знать, кто является отправителем и у взломщика не будет шансов замаскироваться.

—Целостность. Получатель может точно знать, изменялось ли сообщение в процессе доставки, что перекрывает возможность злоумышленнику изменить информацию.

—Неотрицание авторства. Отправитель не сможет ложно отрицать отправку сообщения.

необходимость обеспечения проверки подлинности, целостности и неотрицания авторства – это обязательные требования при общении при помощи компьютеров.

#### 1.1.4 Алгоритмы и ключи

Криптографический алгоритм, также называемый шифром – это математическая функция, используемая для шифрования и дешифрирования.

Обычно это две связанных функции: одна для шифрования, а другая для дешифрирования. Ограниченным алгоритмом называют алгоритм, при котором безопасность алгоритма основана на сохранении самого алгоритма в тайне.

Ограниченные алгоритмы не допускают качественного контроля или стандартизации. Чтобы злоумышленник не мог использовать алгоритм открытых аппаратных или программных продуктов, группам приходится разрабатывать и реализовывать собственные алгоритмы.

Несмотря на эти основные недостатки ограниченные алгоритмы необычайно популярны для приложений с низким уровнем безопасности. Рядового пользователя обычно не интересует уровень его безопасности и общая степень защиты.

В современной **криптографии** главным решением является **криптографический ключ** K. Такой ключ может быть выбран из бесконечного множества значений из заданного промежутка. Множество возможных ключей называют пространством ключей. В процессе шифровки и шифрование, и дешифрование зависят от этого ключа K, и теперь эти функции выглядят так:

, (1)

где M —открытый текст, Е — шифрующая функция, K —секретный ключи С — шифртекст.

, (2)

где M — открытый текст, D — дешифрующая функция, K — секретный ключи С — шифртекст.

При этом выполняется следующее равенство:

, (3)

Где M — открытый текст, D — дешифрующая функция, K — секретный ключ, Е — шифрующая функция.

Для некоторых алгоритмов при шифровании и дешифрировании используются различные ключи. То есть ключ шифрования, К1, отличается от соответствующего ключа дешифрирования, K2.

Так как алгоритм всеобще известен, нарушитель может свободно просматривать алгоритмы шифрования, однако в данной ситуации безопасность зависит именно от персонального ключа. Продукты, использующие такие алгоритмы, могут широко тиражироваться. Пока злоумышленник не знает ключа, передающейся информации ничего не грозит.

**Криптосистема** представляет собой: алгоритм, всевозможные открытые тексты, ключи и шифртексты.

****

Рисунок 1.1.2 – Шифрование и дешифрование с двумя различными ключами

### 1.2 Симметричные алгоритмы

Существует два типа криптографических алгоритмов: и асимметричные. Симметричные алгоритмы (условные) представляют собой алгоритмы, которые дают возможность расшифровать ключ шифрования через ключ дешифрования и наоборот. Чаще всего эти ключи одинковые. Вся безопасность этого алгоритма зависит от ключа. Получив ключ, мошенник получает полный доступ.

**Симметричные алгоритмы** делятся на две категории.

Одни алгоритмы обрабатывают открытый текст побитно и называются потоковыми алгоритмами или потоковыми шифрами.

Другие работают с группами битов (блоками) открытого текста и называются блочными алгоритмами или блочными шифрами.

Для алгоритмов, используемых в компьютерных модемах, типичный размер блока составляет 64 бита -наиболее удобное для работы решение.

### 1.3 Алгоритмы с открытым ключом

Существуют алгоритмы с открытым ключом. Они устроены таким образом, что ключ шифрования и ключ дешифрования различны. Также ключ дешифрования не может быть просчитан по ключу шифрования. Ключ шифрования доступен любому пользователю, однако для дешифрованания предназначен персональный. Более точно, имеют место преобразования:

сообщение + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ШИФРТЕКСТ

ШИФРТЕКСТ + секретный ключ Алисы = сообщение.

Таким образом, Любой может послать Алисе зашифрованную информацию, но только сама Алиса сможет узнать, что было зашифровано в итоге.

### 1.4 Криптоанализ

**Криптоанализ** – это наука получения исходной информации из шифртекста, путем подбора ключей. При успешном криптоанализе, возможно подобрать ключ и получить зашифрованную информацию. Также это дает возможность проверить уязвимости криптосистем. Несанкционированные действия с ключами называется компрометацией ключа.

Все главные положения криптоанализа были описаны в девятнадцатом веке Огюстом Керкгоффс. Основная мысль состоит в том, что ключ – главная системообразующая безопасности.

На данный момент существует четыре основных типа криптоаналитического вскрытия:

1. Вскрытие с использованием только шифртекста.

Для этого случая в распоряжении имеются шифртексты, которые зашифрованы одинаковым алгоритмом. Основная задача – раскрыть открытый текст сообщений, либо получить сам ключ шифрования.

1. Вскрытие с использованием открытого текста

Более простая, в сравнении с первым типом, задача. **Криптоаналитику** даются не только зашифрованные сообщения, но и открытый текст. Основная задача – узнать ключ.

1. Вскрытие с использованием выбранного открытого текста.

В данном случает **криптоаналитик** может сам выбирать шифруемый текст и, более того, может выбирать на базе предыдущих результатов шифрования выбранных текстов. При вскрытии с использованием выбранного открытого текста **криптоаналитик** мог выбрать для шифрования только один большой блок открытого текста, при адаптивном вскрытии с использованием выбранного открытого текста он может выбрать меньший блок открытого текста, затем выбрать следующий блок, используя результаты первого выбора и т. д.

1. Вскрытие с использованием выбранного шифртекста.

Суть в том, что **криптоаналитик** может выбирать различные шифр тексты и иметь их открытый текст. Основная задача – узнать ключ.

1. Вскрытие с использованием выбранного ключа.

**Криптоаналитик** может выбрать ключ и ему предлагается некоторая информация о взаимосвязи ключей.

1. Бандитский криптоанализ.

Использование угроз, пыток и шантажа для получения ключей. К примеру, взяточничество можно назвать иначе – вскрытие с покупкой ключа. Это наиболее продуктивные методы раскрытия алгоритмов, но и наиболее опасные в исполнении.

Лучшими алгоритмами являются те, которые разрабатывались open source, то есть находились с открытым исходным кодом и проходили проверку множеством криптоаналитиков.

## Раздел 2. Аутентификация пользователей в информационных системах.

### 2.1 Простая аутентификация по паролю.

В современной информационной среде аутентификация имеет довольно широкое применение. Рассмотрим несколько ее типов:

1. Протоколы **простой аутентификации** на основе паролей.

Наиболее уязвимый тип при взаимодействии через канал связи, поскольку пароли представляются в открытом виде без их преобразования, но несмотря на это парольная аутентификация достаточно распространена в Интернет, поскольку дает удобный механизм разграничения доступа к удаленным ресурсам без использования защищенных каналов связи и без привлечения доверенных сторон.

1. Протоколы **строгой аутентификации**.

Обеспечивает аутентификацию удаленных пользователей без явного разглашения секретного ключа.

1. Протоколы с **нулевым разглашением**.

Протоколы данного типа основанны на предварительном распределении открытых ключей и их аутентификации; каждый из взаимодействующих субъектов убеждается в подлинности другого субъекта с использованием открытого ключа последнего, причем после процедуры аутентификации никакой из субъектов не получает новой информации о секретном ключе субъекта, подлинность которого подтверждена в ходе осуществления протокола.

Частным вариантом использования процедуры простой аутентификации является парольная защита входа в компьютерную систему. Например, пользователь формирует некоторую случайную информацию и, сохраняя ее в секрете, использует как пароль. Пароль в явном виде не хранится в памяти ЭВМ (или другого устройства, применяемого для выполнения аутентификации). Это требование направлено на то, чтобы потенциальный внутренний нарушитель не имел возможности извлечь из машинной памяти чужой пароль и присвоить себе полномочия другого пользователя.

### 2.2 Протокол рукопожатия

Данный протокол относится к протоколам строгой аутентификации. В действительности «протокол рукопожатия» – это общее название ряда интерактивных протоколов строгой взаимной аутентификации двух пользователей, в которых используется механизм запроса−ответа, с помощью которого каждая из взаимодействующих сторон убеждается, что другая сторона знает некоторый секрет K. Схемы аутентификации такого типа называются протоколами взаимной аутентификации, или протоколами рукопожатия. В качестве секрета может служить разделяемый сторонами общий секретный ключ или личный секретный ключ, связанный с открытым ключом стороны, подлинность которой проверяется. В первом случае ключ доставляется от одной стороны к другой по защищенному каналу или через доверенное лицо.

#### 2.2.1 Протокол с использованием блочных шифров

При использовании некоторого 64-битового блочного шифра E и разделяемого секретного ключа K взаимная аутентификация абонентов А и В осуществляется следующим образом:

1. Абонент В генерирует некоторое случайное число R произвольной длины и посылает его абоненту А.

2. Абонент А, используя заранее оговоренный алгоритм блочного шифрования, шифрует полученное случайное число R по ключу K. Затем он генерирует свое случайное число R′ и пересылает значения R′ и C абоненту В.

3. Абонент В самостоятельно осуществляет такое же преобразование числа R. Если C\* = C, то абонент В принимает решение, что абонент А является подлинным. Затем абонент В шифрует полученное случайное число R′ по ключу K и пересылает значение C′ абоненту А.

#### 2.2.3 Протокол с использованием цифровых подписей.

Пусть абоненты А и В обменялись цифровыми сертификатами, в которых указаны их открытые ключи y А и y В, соответственно, и договорились об использовании некоторой схемы цифровой подписи. Их взаимная аутентификация может быть осуществляется следующим образом.

1. Абонент В генерирует некоторое случайное число R произвольной длины и посылает его абоненту А.

2. Абонент А, используя свой секретный ключ вычисляет значение своей цифровой подписи SА(R) к запросу R. Затем он генерирует свой случайный запрос R′ и пересылает значения R′ и SА(R) абоненту В.

3. Абонент В проверяет подлинность подписи SА(R). Если подпись SА(R) является подлинной, то абонент В принимает решение о подлинности абонента А. Затем абонент В вычисляет значение своей подписи SВ(R′) к R′ и отправляет значение SВ(R′) абоненту А.

#### 2.2.4 Протокол с использованием открытого шифрования

Пусть абоненты А и В обменялись цифровыми сертификатами, в которых указаны их открытые ключи () и (), действующие в рамках криптосхемы RSA. Протокол рукопожатия реализуется следующим образом.

1. Абонент В генерирует некоторое случайное число , зашифровывает его по открытому ключу (). Затем посылает шифртекст C абоненту А.

2. Абонент А, используя свой личный секретный ключ , расшифровывает полученный шифртекст C и получает значение . Затем он генерирует свой случайный запрос R′, зашифровывает его по открытому ключу (): и пересылает значения C′ и R\* абоненту В.

3. Абонент В проверяет выполнимость равенства R\* = R. Если оно, верно, то, используя свой секретный ключ , абонент В расшифровывает полученный шифртекст C′ и получает значение , которое направляет абоненту А.

Протокол рукопожатия позволяет установить взаимную подлинность двух взаимодействующих субъектов. Дело в том, что они могут в дальнейшем отказаться от своих открытых ключей и третья сторона не сможет выявить, кто из этих двух субъектов является нарушителем. Кроме того, нужно оценить потенциальную атаку, состоящую в том, что нарушитель может направлять ложные сообщения сразу после того, как процедура взаимной аутентификации выполнена успешно. Это типичный случай для атак типа «нарушитель посередине». Для противодействия таким атакам следует каждое сообщение сопровождать цифровой подписью отправителя, а получателю – проверять подлинность подписи к каждому полученному сообщению.

## Раздел 3. Протоколы с нулевым разглашением секрета.

### 3.1 Толкование понятия «нулевое разглашение секрета»

При изучении протоколов с нулевыми знаниями важным методическим аспектом является разъяснение термина «**нулевое разглашение секрета**». С понятием нулевого разглашения секрета можно связать понятие стойкости протокола к атакам различного типа. Стойкость криптографических схем одним из центральных понятий в криптографии. При этом для симметричных криптосистем различают практическую **теоретическую стойкость**.

Практическая стойкость связана с тем, что имеющиеся в настоящее время вычислительные ресурсы не позволяют за обозримое время взломать криптосистему, тогда как это принципиально возможно.   
 Теоретическая стойкость предполагает стойкость криптосистемы при наличии у атакующего бесконечных вычислительных ресурсов. По своей природе двухключевые криптосхемы не могут обладать теоретической стойкостью и относятся к классу практически стойких криптосхем.

Для **двухключевых** криптосхем используется понятие доказуемой стойкости. **Двухключевая** криптосхема называется доказуемо стойкой, если можно формально доказать, что взлом криптосхемы не проще, чем решение вычислительно трудной задачи, лежащей в ее основе.

Рассмотрим вопрос о том, когда можно считать, что владелец открытого ключа, по которому выполняется процедура его аутентификации, в ходе выполнения протокола не передает никаких знаний о секрете проверяющей стороне. То, что передаваемые доказывающим значения не содержат в себе информации о секретном ключе, понимается в том смысле, что эти значения не упрощают для потенциального нарушителя задачи вычисления секретного ключа по связанному с ним открытому ключу. Это имеет место в следующих случаях:

1. Доказывающий передает проверяющему значение, которое заведомо известно последнему.
2. Доказывающий передает проверяющему значение, которое последний может вычислить самостоятельно до получения ответа.
3. Доказывающий передает проверяющему случайные значения.
4. Доказывающий передает проверяющему случайные пары или наборы случайных значений, удовлетворяющих некоторому проверочному соотношению. Однако, статистически неразличимые пары или наборы случайных значений могут быть выработаны нарушителем, используя только открытый ключ доказывающего.

### 3.2 Многораундовые протоколы

**Многораундовые** протоколы с нулевым разглашением включают многократное повторения трех типовых шагов: генерация доказывающим, генерация проверяющим и зачислением, доказывающим ответа.

Рассмотрим некоторые из данных протоколов.

#### 3.2.1 Трехшаговый вариант протокола Фиата−Шамира.

Личный секретный ключ каждого пользователя представляет собой два больших сильных простых числа p и q и z последовательных значений (). Открытый ключ вычисляется в виде z упорядоченных значений () по формуле . Вычислительная трудность извлечения квадратного корня по составному модулю n имеет один порядок с вычислительной трудностью задачи разложения модуля n на простые множители, поэтому можно говорить, что описываемый далее протокол основан на вычислительной трудности задачи факторизации. Процедура проверки подлинности владельца открытого ключа, играющего роль доказывающего, выполняется за следующие три шага:

1. Доказывающий выбирает случайное число k, такое, что 1≤k≤n – 1, вычисляет значение u= mod n и посылает его проверяющему.
2. Проверяющий отправляет доказывающему случайную равновероятную z-битовую строку R = () в качестве своего запроса, в которой каждый бит с вероятностью 0,5 равен 1 и с вероятностью 0,5 равен нулю
3. Доказывающий вычисляет свой ответ в виде значения и направляет его проверяющему.

Проверяющий считает ответ положительным, если выполняется соотношение .

#### 3.2.2 Трехшаговый протокол на основе сложности дискретного логарифмирования.

Аутентификация пользователей выполняется по открытому ключу, представляющему собой набор из z значений, где i = 1,2,..., z; α - число достаточно большого простого порядка q по модулю p. Секретным ключом пользователя является набор значений .Протокол состоит из выполнения следующих трех шагов:

1. Доказывающий выбирает текущий разовый секрет k, вычисляет значение (которое играет роль разового открытого ключа) передает число R проверяющему.

2. Проверяющий генерирует свой случайный запрос в виде битовой последовательности E= и направляет ее доказывающему.

3. Доказывающий вычисляет значение ответа на запрос в виде числа и направляет его проверяющему.

Проверяющий считает полученный им ответ правильным, а доказывающего подлинным.

## Раздел 4. Двухшаговые протоколы с нулевым разглашением секрета.

### 4.1 Протоколы на основе алгоритмов открытого шифрования

Построение протоколов с нулевым разглашением можно реализовать, используя известные алгоритмы открытого шифрования. **Для этого** будет использоваться секретный ключ. Для того, кто выполнил процедуру шифрования правильно, восстановленное сообщение будет нести только информацию о том, что тот, кто восстановил зашифрованное сообщение, знает секретный ключ, связанный с открытым ключом.

Однако, потенциальный нарушитель может выбрать произвольное значение и объявить его криптограммой, полученной в результате шифрования сообщения M, и попросить владельца открытого ключа выполнить процедуру расшифрования криптограммы. Если последний это сделает и раскроет восстановленное сообщение, то уже потенциально может иметь место утечка информации о секретном ключе. Говорить, что сообщение не несет в себе информацию о секретном ключе нельзя, так как была раскрыта информация. Поэтому, при построении протоколов с нулевым разглашением секрета, нужен некоторый механизм, который позволяет владельцу открытого ключа (доказывающему) убедиться до раскрытия восстановленного сообщения в том, что последнее уже известно (проверяющему).

В качестве такого механизма могут использоваться алгоритмы **хэширования** (хэш-функции) или заранее специфицированные метки, встраиваемые в исходные сообщения. В протоколах такого типа предполагается использование некоторого алгоритма открытого шифрования Publ\_Encr. Проверяющий может направить доказывающему некоторое случайное сообщение M, предварительно зашифровав его.

Двухшаговый протокол с нулевым разглашением секрета (с нулевыми знаниями) включает следующие шаги:

1. Проверяющий выбирает произвольное сообщение M и, используя специфицированный алгоритм открытого шифрования Publ\_Encr и открытый ключ P доказывающего, зашифровывает сообщение М в шифртекст С=Publ\_Encr(M,P). затем, используя специфицированную хэш-функцию FH, вычисляет значение хэш-функции от М:

H = (M). После этого он отправляет доказывающему пару значений (C, H) в качестве своего запроса.

2. Доказывающий расшифровывает криптограмму С, используя свой личный секретный ключ, в результате чего получает восстановленное сообщение М′. За тем он вычисляет значение хэш-функции от М′: H′ = (M′), сравнивает значения H′ и H и, если H′ = H, то отправляет проверяющему значение M′ в качестве своего ответа. Получив ответ M′, проверяющий сравнивает M ′и M. Если M′ = M, то он делает вывод о подлинности доказывающего.

### 4.2 Протокол на основе использования криптосхемы RSA

Рассмотрим случай построения протокола с нулевым разглашением с использованием алгоритма открытого шифрования RSA, в котором открытый ключ формируется в виде пары чисел (n, e). Первое число представляет собой произведение двух сильных простых чисел rи q, генерируемых по случайному закону, а второе число представляет собой 32-битовое значение, которое выбирается таким, чтобы оно было взаимно простым с числом L(n) = НОК[r − 1, q −1], где НОК −наименьшее общее кратное чисел r −1 и q − 1; L(n) −значение обобщенной функции Эйлера от числа n.

Рассмотрим двухпроходный протокол с нулевым разглашением на основе данного алгоритма открытого шифрования при использовании 128-битовой метки μ = n mod . В качестве метки берутся 128 младших битов открытого ключа доказывающего. Протокол описывается следующими шагами:

1. Проверяющий генерирует случайное сообщение M размером |M|, удовлетворяющим условию |n|/2 <|M| <|n|−|μ|. Затем он зашифровывает сообщение M с присоединенной к нему меткой m, т.е. зашифровывает битовую строку M||m, по открытому ключу доказывающего (n, e), т.е. по формуле С= mod n и направляет доказывающему значение C в качестве своего запроса, на который он ожидает ответ доказывающего.

2. Доказывающий расшифровывает криптограмму С по своему личному секретному ключу dпо формуле M′||μ′ = mod n, где m′−битовая строка, заданная младшими 256 битами расшифрованного значения, и проверяет выполнимость равенства μ′=μ. Если μ′=μ, то полученное значение M′доказывающий направляет проверяющему в качестве своего ответа на полученный запрос. В противном случае доказывающий отправляет ответ «Некорректный запрос».

Если проверяющий получил в качестве ответа правильное значение M′ = M, т. е. то значение, которое он сгенерировал на первом шаге протокола (т. е. до направления своего запроса доказывающему), то делается вывод о подлинности доказывающего.

### 4.3 Использование алгоритма открытого шифрования Эль-Гамаля

Рассмотрим реализацию прокола с нулевым разглашением секрета, основанном на использовании алгоритма открытого шифрования **Эль-Гамаля** . В алгоритме используются открытый ключ вида y = mod p, где p – большое простое число (размером 1024 бит и более), такое, что разложение числа p – 1 содержит простой делитель разрядностью не менее160 бит; α – примитивный элемент по модулю p.

Этот способ фактически представляет собой гибридную криптосистему, в которой секретные ключи распределяются в соответствии с протоколом **Диффи–Хеллмана**, а шифрование сообщения выполняется путем модульного умножения сообщения на секретный ключ. Шифрование сообщения T, отправляемого владельцу открытого ключа y, осуществляется с помощью следующего алгоритма:

1. Сгенерировать случайное число k, которое по своей сути является разовым секретным ключом отправителя сообщения.

2. Вычислить число R= mod p – разовый открытый ключ отправителя.

3. Используя открытый ключ получателя y, вычислить разовый общий секретный ключ Q = mod p.

4. Зашифровать сообщение M путем умножения сообщения на разовый секретный ключ: С= mod p.

5. Отправить получателю криптограмму в виде пары чисел (R, C).

Процедуру открытого шифрования по открытому ключу y обозначим как Gamal\_Encr(M, y). Получатель криптограммы (R, C), используя свой личный секретный ключ x, выполняет процедуру расшифрования Gamal\_Decr(R, C, x), которая описывается следующими шагами:

1. Вычислить разовый общий **секретный ключ** Q= mod p.

2. Используя расширенный алгоритм Евклида, вычислить значение , обратное значению Q по модулю p.

3. Расшифровать сообщение M путем умножения значения C на целое число : M = С mod p.

Если проверяющий не имеет своей целью получение информации о секретном ключе доказывающего, то он корректно выполняет первый шаг протокола с использованием специфицированной метки μ, которую он присоединит к выбираемому им сообщению M, а потом выполнит процедуру зашифрования. В этом случае при расшифровании шифртекста доказывающий восстановит исходное сообщение, увидит в нем специфицированную метку, отбросит ее и получит значение M′ = M.

Проверяющий получает от доказывающего известное ему значение и делает вывод о подлинности доказывающего. Вероятность выполнения соотношения μ′=μ, для произвольно выбранного запроса, пренебрежимо мала и равна значению .

### 4.4 Протокол на основе криптосхемы Рабина.

В алгоритме открытого шифрования **Рабина** используются вычисления по модулю вида n =pq, который используется в качестве открытого ключа. Сильные простые числа p и q составляют личный секретный ключ и удовлетворяют условиям: p≡3 mod 4 и q≡3 mod 4.

Последние два условия обеспечивает снижение сложности процедуры расшифрования криптограммы по сравнению со случаем выбора значений p и q, которые не удовлетворяют указанным двум условиям.

Двухпроходный протокол с нулевым разглашением на основе алгоритма открытого шифрования **Рабина** использует 128-битовую метку m, в качестве которой берутся младшие 128 бит открытого ключа доказывающего, т. е. μ = n mod 2128, и имеет следующий вид:

1. Проверяющий генерирует произвольное сообщение M размером |M|, удовлетворяющим условию |n|/3 < |M| < |n|−|2 μ |, и зашифровывает сообщение M с присоединенной к нему меткой m, т.е. значение M|| μ, по формуле С= (M|| μ) 2 mod n и направляет доказывающему значение C в качестве своего запроса.

2. Доказывающий расшифровывает криптограмму С по своему секретному ключу(p, q) путем вычисления четырех значений квадратного корня из С: M1, M2, M3и M4. Каждое из последних значений он представляет в виде Mi′|| μ i′, где m i′−битовая строка, заданная младшими 128 битами i-го корня (i = 1,2,3,4). Затем он проверяет, выполняется ли равенство μ i′ = μ для одного из четырех значений i. Если да, то соответствующее значение Mi′ доказывающий направляет проверяющему в качестве своего ответа на полученный запрос. Если μ i′≠ μ для i = 1,2,3,4, то доказывающий отправляет ответ «Некорректный запрос».

Использование заранее специфицированной метки позволяет доказывающему идентифицировать исходное сообщение, благодаря чему он направляет проверяющему именно то значение, которое известно проверяющему. Это обеспечивает нулевую утечку информации о секретном ключе при передаче доказывающим своего ответа проверяющему.

## Раздел 5. Протокол на основе схемы открытого согласования ключа

### 5.1 Реализация с использованием криптосхемы Диффи-Хеллмана

Стойкие и удобные в использовании протоколы строгой аутентификации удаленных пользователей могут быть построены **на основе** использования схем открытого согласования ключей. Рассмотрим вариант реализации с использованием схемы **Диффи−Хеллмана,** в которой системными параметрами являются большое простое число p и примитивный элемент α по модулю p. В этой схеме каждый пользователь выбирает случайный секретный ключ x и вычисляет открытый ключ y по формуле y= mod p.

Открытый ключ делается общеизвестным и любой желающий имеет принципиальную возможность однозначно вычислить значение секретного ключа x, хотя эта возможность практически нереализуема, если число p имеет размер, например, не менее 3072 бит, а число содержит простой делитель q размером не менее 256 бит. Рассмотрим протокол аутентификации удаленных пользователей с нулевым разглашением, который использует значение α, имеющее по модулю p порядок, равный q, и включает следующие два шага:

1. Проверяющий генерирует случайное число k и вычисляет значение своего разового открытого ключа U=и значение Z=где y−открытый ключ доказывающего (т.е. пользователя, подлинность которого проверяется), после чего передает доказывающему значение U в качестве своего запроса, на который он ожидает ответ доказывающего.

2. Доказывающий проверяет выполнимость соотношения Если да, то он вычисляет значение Z=и направляет Z проверяющему в качестве своего ответа на полученный запрос. Если указанное соотношение не выполняется, то доказывающий направляет ответ «Некорректный запрос».

### 5.2 Реализация с использованием простых чисел Мерсенна

**В качестве** циклических групп простого порядка можно использовать мультипликативные группы конечных полей GF() двоичных многочленов для некоторых значений степени s. Порядок мультипликативной группы таких полей равен q = −1. Значение −1 является простым, если значение s равно степени **Мерсенна**. (**Степень Мерсенна** – это такое простое число s, при котором значение −1 тоже является простым числом.)

Данные многочлены представляют интерес для использования при реализации протоколов аутентификации с нулевым разглашением секрета на основе схем открытого согласования ключа.

Таблица 1- **Неприводимые** двоичные многочлены малого веса.

|  |  |
| --- | --- |
| Степень Мерсенна | Двоичные многочлены |
| 1279 |  |
| 2203 |  |
| 2281 |  |
| 3217 |  |
| 4423 |  |
| 9689 |  |
| 9941 |  |
| 11213 |  |
| 19937 |  |

Практический интерес представляет реализация этого протокола с использованием идеальных эллиптических кривых (все точки которых, кроме бесконечно удаленной точки, имеют одно и то же значение порядка, равное простому числу) вместо конечных мультипликативных групп двоичных многочленов. Такая замена используемой конечной группы простого порядка обеспечивает существенное уменьшение вычислительной сложности протокола при заданном уровне стойкости.

## Раздел 6. Протокол на основе последовательного возведения в квадрат

### 6.1 Протокол на основе последовательного возведения в квадрат

Рассмотрим **двухпроходный протокол** с нулевым разглашением, основанный на трудности задачи факторизации, для которого доказательство нулевой утечки секрета является достаточно очевидным. В качестве открытого ключа доказывающего используется натуральное число n, равное произведению двух больших простых чисел r и q, составляющих его личный секрет.

Идея доказательства отсутствия утечки информации о секретном ключе состоит в том, что доказывающий передает проверяющему значение, которое вычислено до того, как оно было вычислено доказывающим, поэтому никакой новой информации проверяющему не передается. Протокол включает следующие два шага:

1. Проверяющий генерирует случайное число a <n и ρ-битовое число k, где ρ = (значение ρ выбирается с учетом времени задержки δ, вносимого каналом связи). Затем используя метод последовательного возведения в квадрат, вычисляет значение, T= после чего передает доказывающему пару значений a и k в качестве своего запроса, на который он ожидает ответ доказывающего;

2. Доказывающий выполняет две последовательные операции возведения в степень, в результате чего за короткое время вычисляет значения K = mod L(n), где L(n) −обобщенная функция Эйлера от числа n, и T = mod n. Затем он сразу направляет проверяющему значение T в качестве своего ответа на полученный запрос.

Без знания факторизации модуля вычисление значения T= выполняется методом последовательного возведения в квадрат. При этом требуется выполнить k возведений в квадрат (умножений) по модулю n.

### 6.2 Особенности практического применения

Практическое применение протокола аутентификации, описанного в предыдущем подразделе, требует учета возможных вычислительных ресурсов у потенциального нарушителя. Если предполагается возможность применения нарушителем специализированных производительных ЭВМ (применение многопроцессорных ЭВМ не дает эффекта, так как процесс последовательного возведения в квадрат не может быть распараллелен), то проверяющему требуется выбирать более длинные числа k. При этом ему потребуется потратить больше времени на вычисление значения T.

Выбираемая разрядность k определяется также и быстродействием канала связи, используемого в протоколе. Чем более быстродействующий канал, тем меньшее значение ∆ может быть выбрано, т. е. тем меньшая разрядность числа k может быть использована. Это позволит уменьшить время вычислений, выполняемых проверяющим до направления запроса доказывающему.

При практическом использовании этого протокола может быть применен способ **предвычислений**, который состоит в том, что проверяющий заблаговременно вычисляет множество значений (a, k, T), которые впоследствии будут использоваться при осуществлении аутентификации удаленных пользователей.

Следует заметить, что использование технического параметра канала связи, связанного с быстродействием, вносит существенные ограничения на область применения данного протокола.

## **Раздел 7. Преобразование протоколов с нулевым разглашением в схемы** цифровой подписи

### 7.1 Общий подход к выводу схем цифровой подписи и протоколов с нулевым разглашением

Протоколы с использованием **фиксатора** включают следующие три шага:

1. Генерация доказывающим значения фиксатора и направление его проверяющему.

2. Генерация проверяющим случайного запроса и отправка его доказывающему.

3. Вычисление доказывающим ответа на запрос и передача ответа проверяющему.

Правильность ответа проверяется по некоторому проверочному соотношению. В случае многораундовых протоколов эти три шага выполняются многократно. Любой протокол с нулевым разглашением, который можно отнести к этому типу, может быть преобразован в схему цифровой подписи.

В соответствии с протоколом с нулевым разглашением до формирования запроса доказывающий формирует значение одного или z различных фиксаторов и после получения запроса доказывающий не может поменять значение фиксатора. Поэтому, рассматривая некоторый документ как источник случайного запроса, нужно использовать такой механизм формирования запроса, который требует использования значения фиксатора, т. е. запрос должен формироваться не только в зависимости от каждого бита документа, но и в зависимости от каждого бита фиксатора.

Схемы ЭЦП, построенные с использованием преобразования протокола с нулевым разглашением, можно отнести к доказуемо стойким схемам цифровой подписи, поскольку предположение о нестойкости схемы цифровой подписи, использующей стойкую хэш-функцию, влечет за собой вывод о нестойкости протокола с нулевым разглашением, из которого была выведена схема цифровой подписи.

### 7.2 Схема цифровой подписи Фиата-Шамира

Эта схема цифровой подписи основана на сложности вычисления квадратных корней по модулю n, являющимся произведением двух больших простых чисел p и q [14], и ее можно представить как результат преобразования трехшагового протокола с нулевым разглашением **Фиата–Шамира**, описанного в разделе 3.3.

Возможны два варианта:

1. каждый абонент формирует свой индивидуальный модуль n =pq

2. все абоненты используют одно и то же значение составного модуля n, которое вырабатывается доверительным центром, причем исходные простые множители p и q уничтожаются (для формирования открытых и закрытых ключей не требуется знать разложение модуля).

Первый случай является предпочтительным, поскольку может обеспечить гарантии абоненту в том, что разложение модуля неизвестно потенциальному нарушителю.

Достоинством рассмотренной схемы **ЭЦП** является сравнительно малая сложность процедуры формирования и проверки подписи, что обеспечивает достаточно высокое быстродействие. Недостатком является большой размер открытого и закрытого ключей.

### 7.3 Схема цифровой подписи на основе вычислительной трудности задачи дискретного логарифмирования

Рассмотрим преобразование трехшагового протокола с нулевым разглашением, основанного на вычислительной трудности задачи дискретного логарифмирования в схему цифровой подписи. В соответствии с исходным протоколом открытым ключом является упорядоченный набор () из z значений , вычисляемых по формуле, где i = 1,2,..., z; α - число достаточно большого простого порядка q по модулю p. Секретным ключом пользователя является упорядоченный набор () случайных значений .

Генерация цифровой подписи владельцем открытого ключа заключается в выполнении следующих шагов.

1. Подписывающий генерирует разовый секрет k, вычисляет значение фиксатора R = mod p, которое является первым (рандомизирующим) элементом ЭЦП. Затем он объединяет фиксатор и подписываемое сообщение M и вычисляет значение хэш-функции , которое рассматривается как упорядоченная последовательность однобитовых запросов, задаваемых подписываемым документом. (При выполнении процедуры проверки подписи эта битовая цепочка запросов восстанавливается, после чего будет проверяться правильность ответа на нее.)

2. Второй элемент цифровой подписи вычисляется в виде значения S по формуле .

### 7.4 Формальное доказательство стойкости схем цифровой подписи

Тот факт, что схема цифровой подписи **Шнорра** является реализацией протокола с нулевым разглашением секрета с участием подписываемого документа (роль проверяющего может быть выполнена любым лицом, заинтересованным в проверке подлинности подписи) **может быть рассмотрено** как формальное доказательство стойкости схемы.

**На самом деле,** вопрос о доказательстве нулевого разглашения секрета в ходе протокола с нулевым разглашением и доказательство практической (вычислительной) невозможности дать правильный ответ на случайный запрос без знания секретного ключа – это разные аспекты оценки стойкости.

**Известный ранее** способ формального доказательства стойкости этой схемы подписи представляется более громоздким и в нем также используется предположение о том, что в схеме цифровой подписи используется стойкая хэш-функция, поэтому атакующий не может воспользоваться наличием ее слабостей.

Это было первое в мире формальное доказательство стойкости схемы ЭЦП, основанное на вычислительной трудности задачи дискретного логарифмирования, однако оно не может быть распространено на стандарты цифровой подписи России и США из-за отличий в процедуре формирования подписи.

Формальное доказательство стойкости схем цифровой подписи путем их вывода из протоколов с нулевым разглашением, что предложено в работе, представляется более общим подходом, и может быть применено для случая указанных стандартов. Однако, для каждого из них, не в полной мере проработан вопрос об обосновании базового протокола с нулевым разглашением, из которого выводится схема.

## Раздел 8. Снижение сложности вычислений в криптосхемах, основанных на задаче дискретного логарифмирования.

### 8.1 Метод Монтгомери

**Рассмотрим** применение метода Монтгомери в случае модульного умножения многочленов, заданных над конечным простым полем GF(p). Следует отметить, что ускорение операции модульного умножения достигается в случае, когда требуется выполнить достаточно много операций умножения по заданному модулю, что имеет место при выполнении операции возведения в большую степень. Последнее является типичным для криптосхем с открытым ключом, основанных на трудности задачи дискретного логарифмирования в конечной циклической группе и задачи факторизации.

В рассматриваемом методе каждому многочлену (x) ставится в соответствие некоторый образ-многочлен F[m(x)] и вычисление осуществляется над образами, а после получения результата осуществляется переход от образа к многочлену.

**Предположим**, требуется выполнять многократные умножения по модулю многочлена η(x). При этом выбирается некоторый многочлен ρ(x), такой что где deg обозначает степень многочлена. Ускорение операции умножения по модулю η(x) достигается за счет того, что операция деления на η(x) в арифметике Монтгомери заменяется на операцию деления на многочлен ρ(x), в качестве которого выбирается одночлен , где n– натуральное число, и тогда операция деления сводится к записи коэффициентов делимого многочлена, стоящих при степенях переменной, не превосходящих n.

### 8.2 Выбор простых чисел специального вида

Вычислительная сложность модульного умножения чисел существенно зависит от вида числа, используемого в качестве модуля. В криптосхемах в качестве модуля чаще всего используются простые числа, поэтому будем рассматривать случай простого модуля, однако способ ускорения операции модульного умножения, который состоит в выборе модуля, запись которого в некоторой системе счисления содержит малое число ненулевых разрядов, применим также и в случае, когда модуль не является простым числом. Наиболее простой для рассмотрения случай относится к двоичной системе счисления.

### 8.3 Выбор неприводимых многочленов специального вида

Повышение производительности двухключевых криптосхем, построенных с использованием трудности задачи дискретного логарифмирования в конечных полях многочленов, заданных над простым полем GF(p), связано со снижением сложности операции умножения многочленов по модулю неприводимого многочлена. Прямолинейная реализация этой операции связана с выполнением арифметического умножения двух многочленов и арифметическим деление полученного результата на заданный в качестве модуля неприводимый многочлен.

В качестве результата модульного умножения берется остаток от указанной операции деления. Остатком является многочлен, степень которого, по крайней мере, на единицу меньше степени многочлена-модуля. При этом временная сложность операции арифметического деления существенно превышает сложность операции арифметического умножения многочленов.

Уменьшение вычислительной сложности операции модульного умножения многочленов может быть достигнуто следующими способами:

1. Заменой деления на заданный модуль-многочлен степени n на деление на одночлен степени n +1 в соответствии с методом **Монтгомери**.
2. Выбором в качестве модуля многочлена с небольшим числом ненулевых коэффициентов.

### 8.4 Генерация неприводимых многочленов

Для генерации неприводимых многочленов общего вида может быть использован следующий алгоритм:

1. Формируем произвольный многочлен f(x) над полем GF(p).

2. Выполняем его проверку на неприводимость по детерминистическому тесту, который рассматривается далее.

3. Если тест на неприводимость показал, что f(x) является неприводимым, то вывести этот многочлен как результат выполнения алгоритма. В противном случае перейти к шагу 1.

**Выход**: неприводимый многочлен f(x).

В случае генерации двоичных многочленов (p=2), степени которых представимы в виде s= (h– натуральное число) имеем такое разложение для шага 1:. Это можно использовать для упрощения компьютерной программы для генерации двоичных многочленов, имеющих указанные значения степени. Для других частных случаев, когда значения p и s заранее известны, все простые делители числа можно вычислить и записать в виде таблицы, к которой будет обращаться компьютерная программа, вместо того, чтобы на шаге 1 выполнять процедуру факторизации числа .

## Вывод

**Таким образом**, в первой главе курсовой работы была разобрана теоретическая часть алгоритмов аутентификации, произведен анализ основных протоколов для шифрования и дешифрования информации.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Молдовян, А. А. Протоколы аутентификации с нулевым разглашением секрета / А. А. Молдовян, Д. Н. Молдовян, А. Б. Левина. — Санкт-Петербург : Университет ИТМО, 2016. — 55 c. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/68058.html (дата обращения: 29.10.2020). — Режим доступа: для авторизир. пользователей

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) – URL: [http://docs.cntd.ru/document/1200157208](https://vk.com/away.php?to=http%3A%2F%2Fdocs.cntd.ru%2Fdocument%2F1200157208&cc_key=) (дата обращения: 29.10.2020). – Текст: электронный.  
3. ГОСТ 19.201-78. Межгосударственный стандарт. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению (введен в действие Постановлением Госстандарта СССР от 18.12.1978 N 3351) (ред. от 01.06.1981) – URL: [http://docs.cntd.ru/document/gost-19-201-78](https://vk.com/away.php?to=http%3A%2F%2Fdocs.cntd.ru%2Fdocument%2Fgost-19-201-78&cc_key=) (дата обращения: 29.10.2020). – Текст: электронный.