Государственное бюджетное общеобразовательное учреждение города Москвы "Школа № 967"

Тематика: Информатика, Обществознание

Создание программного обеспечения для защиты права на тайну переписки

Класс: 10

Автор работы: Кириллов Михаил Александрович

Научный руководитель проекта: Юрченко Людмила Алексеевна

Оглавление

Введение1

Аннотация2

**I Глава. Теоретическая основа3**

Современные мессенджеры и их проблемы3

P2P технологии4

Об устройстве использованной криптографической системы5

Об устройстве технологического решения7

**II Глава. Практическое применение теории8**

Реализация ПО на языке программирования8

Тестирование ПО на наличие ошибок9

Развитие программы PISA10

Вывод11

Список использованной литературы13

Приложения15

Введение

Вашему вниманию представляется проектная работа: “Создание программного обеспечения для защиты права на тайну переписки”.

Наша методология исследования.

В современном мире люди пользуются множеством различных приложений для общения. Большая часть из них принадлежит крупным компаниям, с которыми эти пользователи заключают соглашение при первом запуске программы. В подобных соглашениях часто можно увидеть пункты о продаже обезличенных данных этих людей третьим лицам. Но мало кто в наше время читает лицензионные соглашения. И в этом мы одновременно видим актуальность нашего проекта и большую проблему, которую можно решить.

Мы предполагаем, что возможно создать программное обеспечение, которое будет передавать по сети сообщения, не используя дорогостоящего оборудования, не передавая данные пользователей третьим лицам, а также имеющее открытый исходный код, с которым сможет ознакомиться любой любопытный или желающий создать свою версию приложения человек.

Поэтому целью нашей работы является создание ПО, способного передавать информацию по сети, сохраняя её приватность, не используя дорогое оборудование и способное запускаться на маломощных персональных компьютерах, имеющее открытый исходный код на одной из популярных площадок для хранения оного.

Объект и предмет исследования Вы можете увидеть на слайде.

Для решения указанных задач необходимо было применить следующие методы исследования: теоретический, (который заключался в анализе профессиональной литературы), и эмпирический, (представляющий собой создание и тестирование ПО).

Для достижения поставленной цели необходимо было решить перечень задач, которые представлены на слайде.

На следующем слайде Вы можете увидеть план реализации нашего проекта.

Аннотация

Проект “Создание программного обеспечения для защиты права на тайну переписки” направлен на создание программного обеспечения, способного передавать информацию по сети, сохраняя её приватность, не используя дорогое оборудование и способное запускаться на маломощных персональных компьютерах, имеющее открытый исходный код на одной из популярных площадок для хранения оного.

Тип проекта: Технический

Продукт проекта: Приложение для персональных компьютеров

Данная работа знакомит читателя с основными понятиями, которые использовались при проектировании и создании приложения. Так же в этом проекте рассказывается об этапах написания и тестирования данного программного обеспечения.

Проект имеет инновационный характер, и будет интересен тем, кто беспокоиться о безопасности и свободе передачи информации по сети, а также тем, кто не желает заключать невыгодные соглашения с компаниями, о передаче информации о пользователе третьим лицам. Созданное в ходе проекта приложение может быть использовано и/или усовершенствованно любым желающим, так как его код будет доступен на GitHub.

I Глава

Теоретическая основа

Современные мессенджеры и их проблемы

Сейчас на рынке приложений для смартфонов и ПК существует множество известных и повсеместно используемых мессенджеров. Самые известные – это WhatsApp, Viber, Facebook Messenger, Telegram и Skype.

Для примера возьмём WhatsApp. Если прочесть его лицензионное соглашение[1], то становиться очевидным, что данные, которые собирает о вас компания, а именно: ваш номер телефона, номера телефона ваших контактов, фото вашего профиля, ваш статус, платёжная информация, модель вашего устройства, информация об ОС, данные браузера, IP адрес, ваше местоположение и прочее. Всей этой информацией компания может распоряжаться по своему усмотрению, делясь большей её частью с остальными продуктами Facebook, а меньшей – со сторонними организациями и третьими лицами.

Кроме лицензионных соглашений, все эти мессенджеры имеют хотя бы один недостаток из тех, что вы видите на слайде. WhatsApp, Viber, Facebook Messenger и Skype имеют все три недостатка (централизованная архитектура, закрытый код, закрытый протокол), а Telegram только один (централизованная архитектура).

Решением проблем этих приложений может стать создания мессенджера с децентрализованной архитектурой, открытым кодом, известным протоколом передачи информации и отсутствием необходимости заключения лицензионного соглашения.

Для этого нам необходимо немного погрузиться в мир сетей и криптографии. Начнём с первого.

P2P технологии

Архитектуры сетей можно разделить на две большие группы – одноранговые сети и клиент-серверные сети. Начнём с первых.

P2P или одноранговая сеть[2] – это сеть основанная на равноправии участников, в которой отдельные участники – узлы или пиры – выполняют одновременно функции сервера и клиента (демонстрация принципа на Приложении А).

А клиент-серверная сеть[3] – это сетевая архитектура, в которой присутствуют клиенты, посылающие через сеть, используя сетевые протоколы, запросы на сервера, которые в свою очередь предоставляют им услугу в виде данных или сервисных функций (демонстрация принципа на Приложении Б).

Сразу можно заметить некоторые преимущества P2P сетей над клиент-серверными сетями. Мы бы выделили самые важные для нас:

* Стоимость построения и эксплуатации P2P сети сравнительно ниже, чем у клиент-серверной архитектуры сети, что соответствует нашей цели
* Нет необходимости в сетевом администраторе, что так же говорит об отсутствии необходимости на дополнительные расходы
* Гораздо большая надёжность - выход из строя или отключения одного узла не влияет на всю сеть, что является преимуществом будущего приложения, так как отсутствует возможность проведения Dos или DDos атаки.
* Нет возможности воспрепятствовать распространению информации внутри сети, что обеспечивает свободу слова пользователей

Но имея только хорошую сетевую архитектуру, мы не можем создать приложение. Нам нужно что-то, что будет сохранять приватность сообщений наших пользователей. Для этого нам нужна криптографическая система.

Об устройстве использованной криптографической системы

В этой главе я расскажу о теории, которую мы изучили, а также обосную наш выбор криптографической системы для приложения.

Существует два вида криптографических систем, с возможностью расшифровать зашифрованное сообщение. Это симметричное и ассиметричное шифрование[4].

В симметричной криптографической системе используется только один ключ для шифрования и дешифрования сообщения. Так же к этому виду систем относятся и те, в которых имеется два ключа, но имея один из ключей, сложность вычисления второго можно оценить как лёгкую. В таких системах безопасность основывается на том, что ключ для шифрования и дешифрования известен только получателю и отправителю сообщения[5][6]

В такой системе необходимо решить задачу передачи ключа шифрования таким образом, чтобы этот ключ не попал в руки к злоумышленнику.

(Пример работы криптографической системы на Приложении В)

Вместо решения этой задачи можно пойти более простым путём и воспользоваться криптографическими системами с открытым ключом. Или как их ещё называют – Ассиметричные криптографические системы.

В этой криптографической системе используется два ключа – публичный (или открытый) и приватный (или секретный). Имея только один из ключей, сложность вычисления другого можно оценить как очень сложную, так как придётся решить задачу факторизации больших целых чисел (разложения большого целого числа на два простых), на что потребуется очень много времени. И именно этой сложности, и секретности приватного ключа основывается безопасность ассиметричных алгоритмов[7][8].

В такой системе нет необходимости решать задачу отправки публичного ключа таким образом, чтобы его не перехватили, так как публичный ключ может быть у любого человека и это не несёт ущерба безопасности общения.

(Пример работы криптографической системы на Приложении Г)

Из всех имеющихся ассиметричных систем шифрования нами был выбран алгоритм под названием RSA, происходящее от имён создателей. Этот алгоритм весьма популярен и используется в таких приложениях как PGP, TLS/SSL и др. Так же это алгоритм был подвергнут нескольким аудитам безопасности, которые подтвердили его криптостойкость, при использовании ключей достаточной длины.

Осталось только выбрать длину ключа, параметра, от которого зависит криптостойкость алгоритма[9] – вероятность его взлома при криптоанализе. Нами был выбран ключ длиной 2048 бит, так как недавние успехи математиков в алгоритмах факторизации больших целых чисел смогли скомпрометировать алгоритм шифрования RSA с длиной ключа 1024 и меньше[10].

Об устройстве технологического решения

Для удобства работы в практической части проекта, мы составили техническое задание. Вот пункты оттуда, для ознакомления и понимания задач, которые поставлены при создании приложения.

(Техническое задание – Приложение Д)

Принцип работы созданного приложения, с учётом особенностей используемых технологий, а также варианты использования приложения с учётом различных ситуаций использования описан на следующих слайдах, сопровождённых схемами.

(Приложения Е, Ж, З, И)

II Глава

Практическое применение теории

Реализация ПО (PISA) на языке программирования Java

Приложение, соответствующее описанию в техническом задании и прошедшее все тесты приёма, написанное на языке программирования Java, представлено на слайде в виде UML диаграммы, для лучшего восприятия и понимания строения приложения.

(Приложение Й)

У любого приложения есть название, поэтому мы решили, что будет хорошей идеей дать нашему ПО название PISA (ПИЗА), в честь пизанской башни.

Само название является красивой аббревиатурой от Private Information Sharing Application, что можно перевести как “Приложение для обмена частной информацией”.

Тестирование ПО на наличие ошибок

При тестировании ПО на безопасность была выявлена уязвимость, позволявшая подключаться третьему “собеседнику” и прослушивать сообщения жертвы, тогда как изначальный собеседник отключался от общения. Часть кода, в которой была уязвимость, вы можете видеть на слайде.

(Приложение К)

Развитие программы PISA

Перед тем, как перейти к выводу и заключению, нам бы хотелось заметить потенциал развития программы PISA. Из-за ограниченного времени на не удалось его полностью реализовать, но в будущем мы планируем добавить следующее:

* Групповые чаты
* Видео и аудио звонки
* Получение и отправка мультимедиа
* Сохранение переписки в зашифрованном виде
* Возможность запуска приложения с графическим окружением

Вывод

На основе полученных в ходе работы над проектом знаний и практической работы, мы экспериментально доказали, что создать программное обеспечение, способное передавать сообщения по сети, не используя дорогостоящего оборудования, и код которого будет открыт для любого пользователя возможно.

Так же вы можете [ознакомиться](https://github.com/anOnim-creator/PISA) с кодом или [скачать](https://github.com/anOnim-creator/PISA/releases/download/Version_1.0/PISA.jar) приложение по следующим ссылкам или используя QR коды.

(Приложения Л, М)

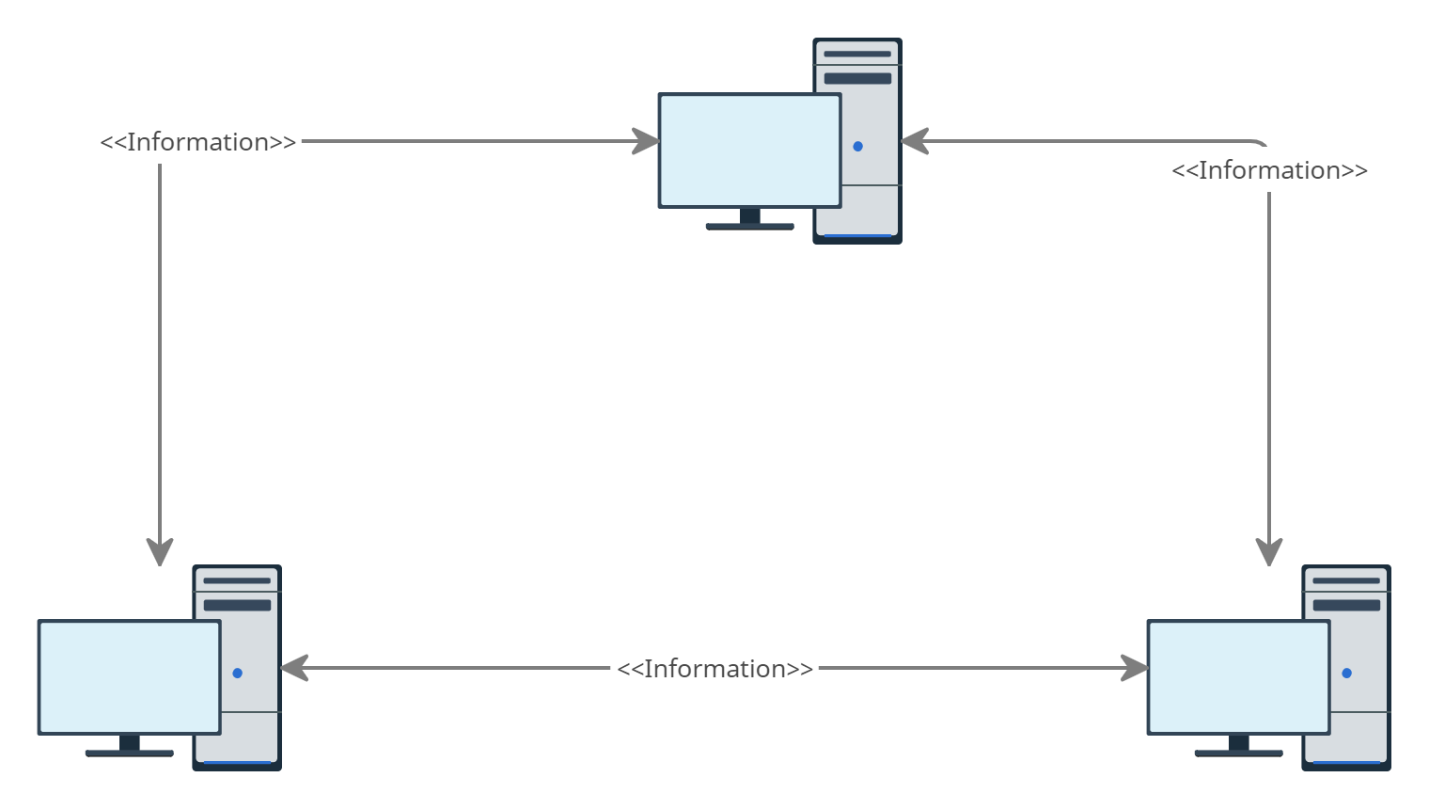
Заключение

Я считаю, что мне удалось решить выявленную мной проблему, подтвердить свою гипотезу и достичь свою цель, используя поставленные задачи и пользуясь известными мне методами.

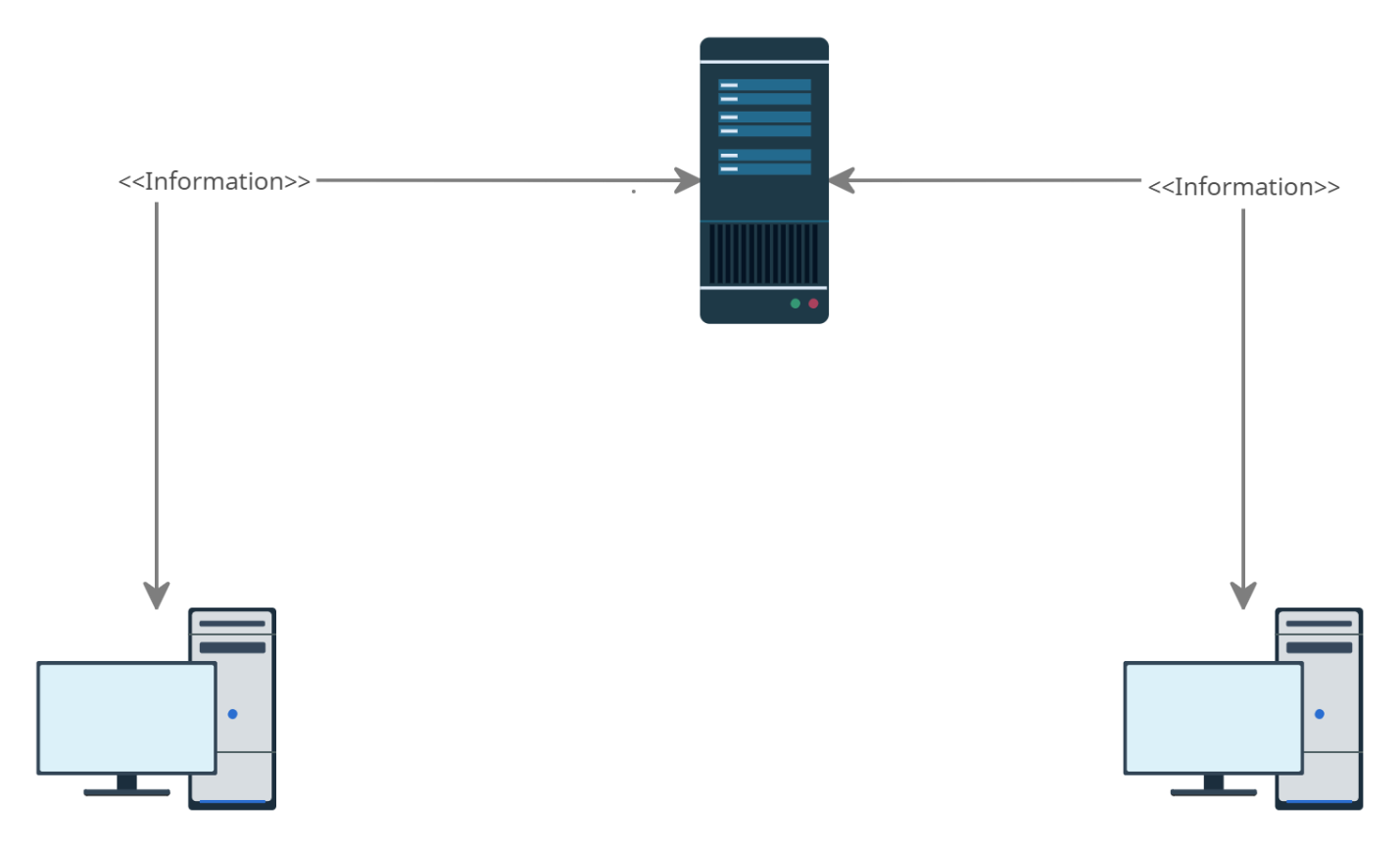
Список использованной литературы

1. Политика конфиденциальности WhatsApp, пункты: “Информация, которую мы собираем”, “Как мы используем информацию”, “Информация, которой делитесь вы и WhatsApp”, “Аффилированные компании”, “Уступка, смена собственников и передача”, “Управление вашей информацией” и “Законодательство и защита” [Электронный ресурс]. – URL: <https://www.whatsapp.com/legal/privacy-policy>
2. Одноранговая сеть [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%80%D0%B0%D0%BD%D0%B3%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C>
3. Клиент-серверная сеть [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_%E2%80%94_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80>
4. Криптография [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F>
5. Симметричные криптосистемы [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B5_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B>
6. [Сообщество IT специалистов Хабр, Введение в криптографию и шифрование, часть первая. Лекция в Яндексе](https://habr.com/ru/company/yandex/blog/324866/)
7. Криптосистема с открытым ключом [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC>
8. [Сообщество IT специалистов Хабр, Введение в криптографию и шифрование, часть вторая. Лекция в Яндексе](https://habr.com/ru/company/yandex/blog/327636/)
9. Ключ (криптография) [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)>
10. Dan Goodin. New crypto-cracking record reached, with less help than usual from Moore’s Law/Dan Goodin // Ars Technica – 2019 [Электронный ресурс]. – URL: <https://arstechnica.com/information-technology/2019/12/new-crypto-cracking-record-reached-with-less-help-than-usual-from-moores-law/>

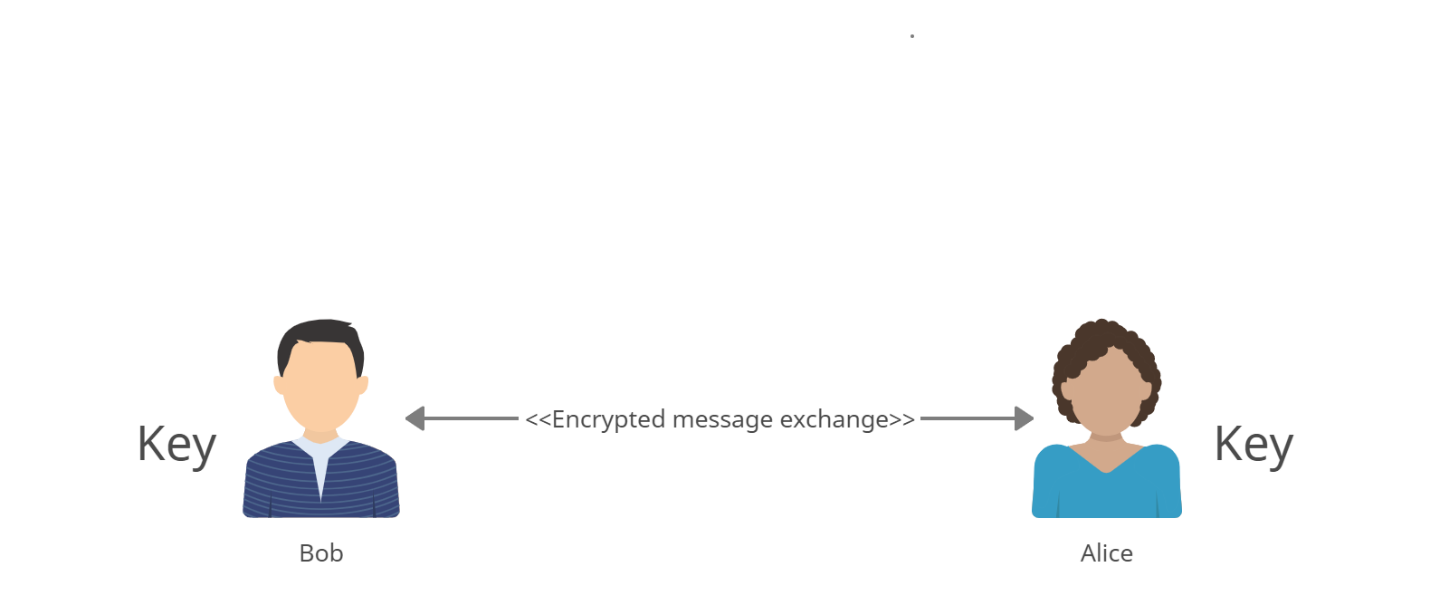
Приложение А



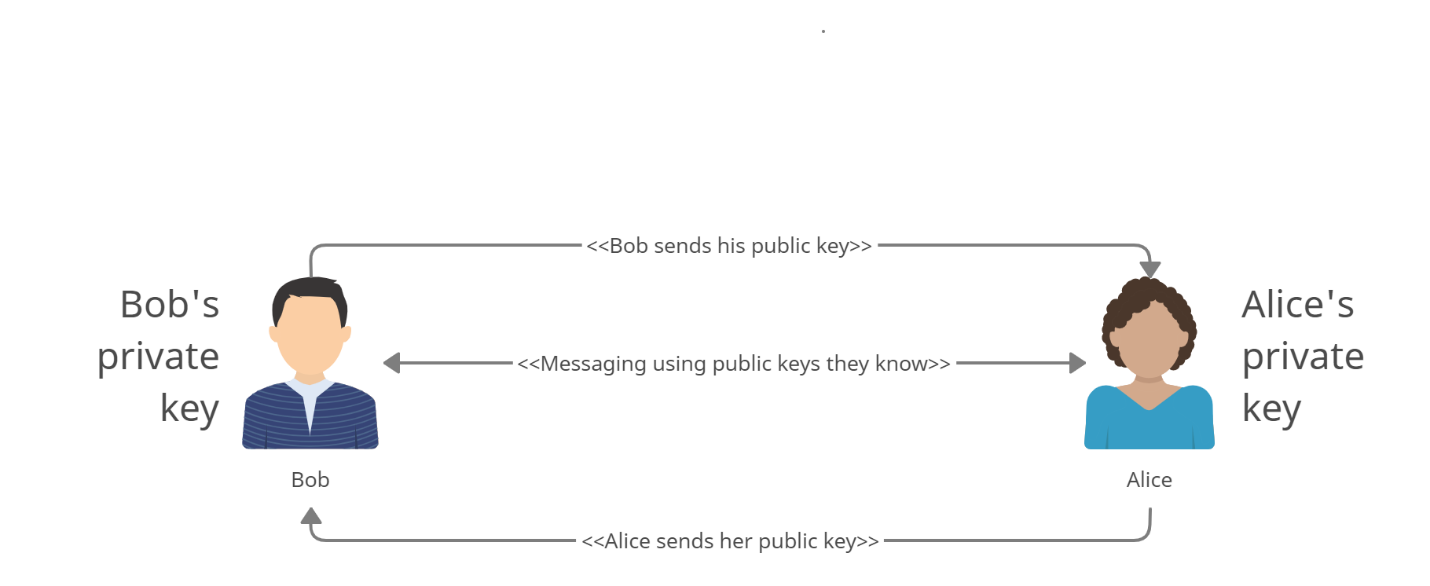
Приложение Б



Приложение В



Приложение Г



Приложение Д

**Техническое задание**

**Задача:**

Создать консольное приложение на языке Java, которое сможет отсылать и принимать зашифрованные сообщения по сети.

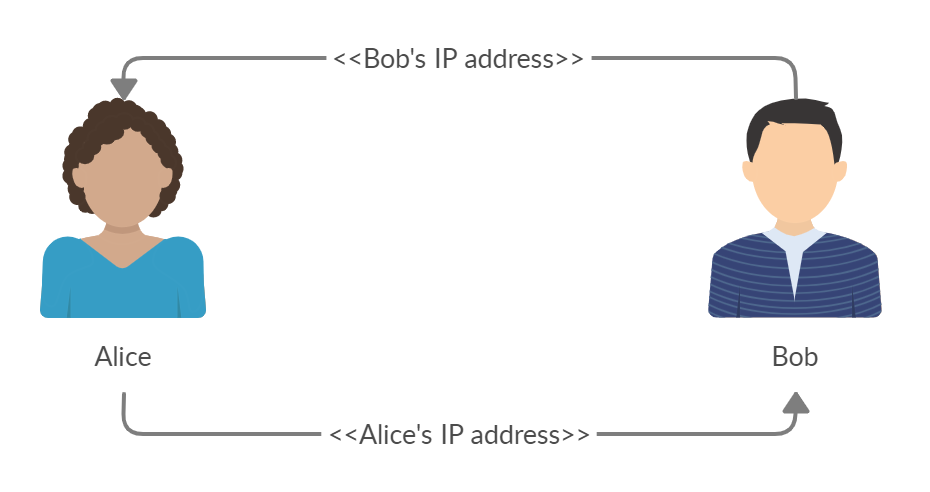
**Описание:**

1. Пользователи, использующие приложение, должны создавать одноранговую сеть.
2. Для определения устройств в сети использовать Internet Protocol Address.
3. Для отправки и приёма сообщений приложение должно использовать потоки. Само сообщение должно быть представлено отдельным классом. Часть приложения, которая принимает сообщения от собеседника, должна работать в фоновом потоке. Часть приложения, отправляющая сообщения пользователя, должна работать в основном потоке.
4. Должен использоваться криптографический алгоритм – RSA, с длиной ключа 2048.
5. Обмен ключами должен происходить при подключении собеседников друг к другу.
6. Перед отправкой сообщение должно быть зашифровано.
7. Необходимо проверять полученное сообщение с хэш-суммой изначально отправленного сообщения, для выявления несанкционированных изменений в сообщение.
8. Срок выполнения – до января 2021 года

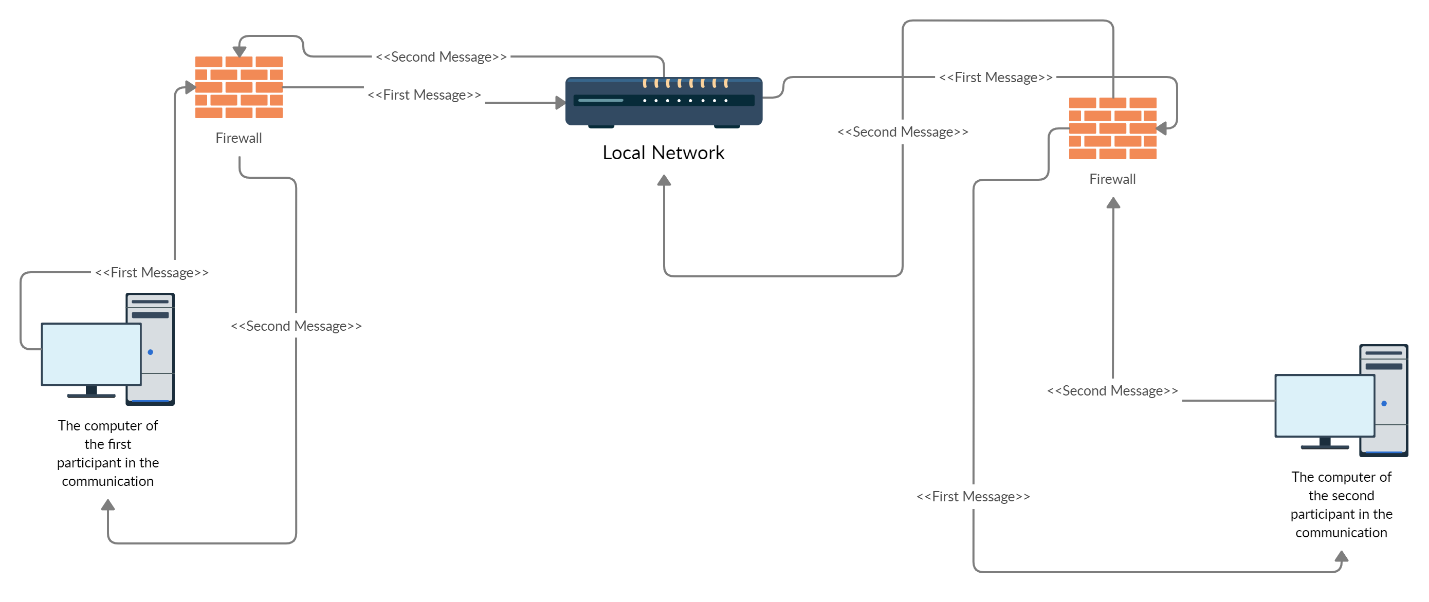
**Правила приёма:**

1. Приложение должно обрабатывать ввод сообщения различной длины.
2. Приложение должно работать на вычислительных машинах с различными вычислительными мощностями.
3. Приложение должно обеспечивать приватность передаваемой информации. После окончания сеанса общения вся полученная информация удаляется.
4. Приложение не обязано корректно работать в случае, если имеющийся IP адрес пользователя не имеет функции открытия порта.
5. Приложение должно работать на операционных системах Windows и Linux.

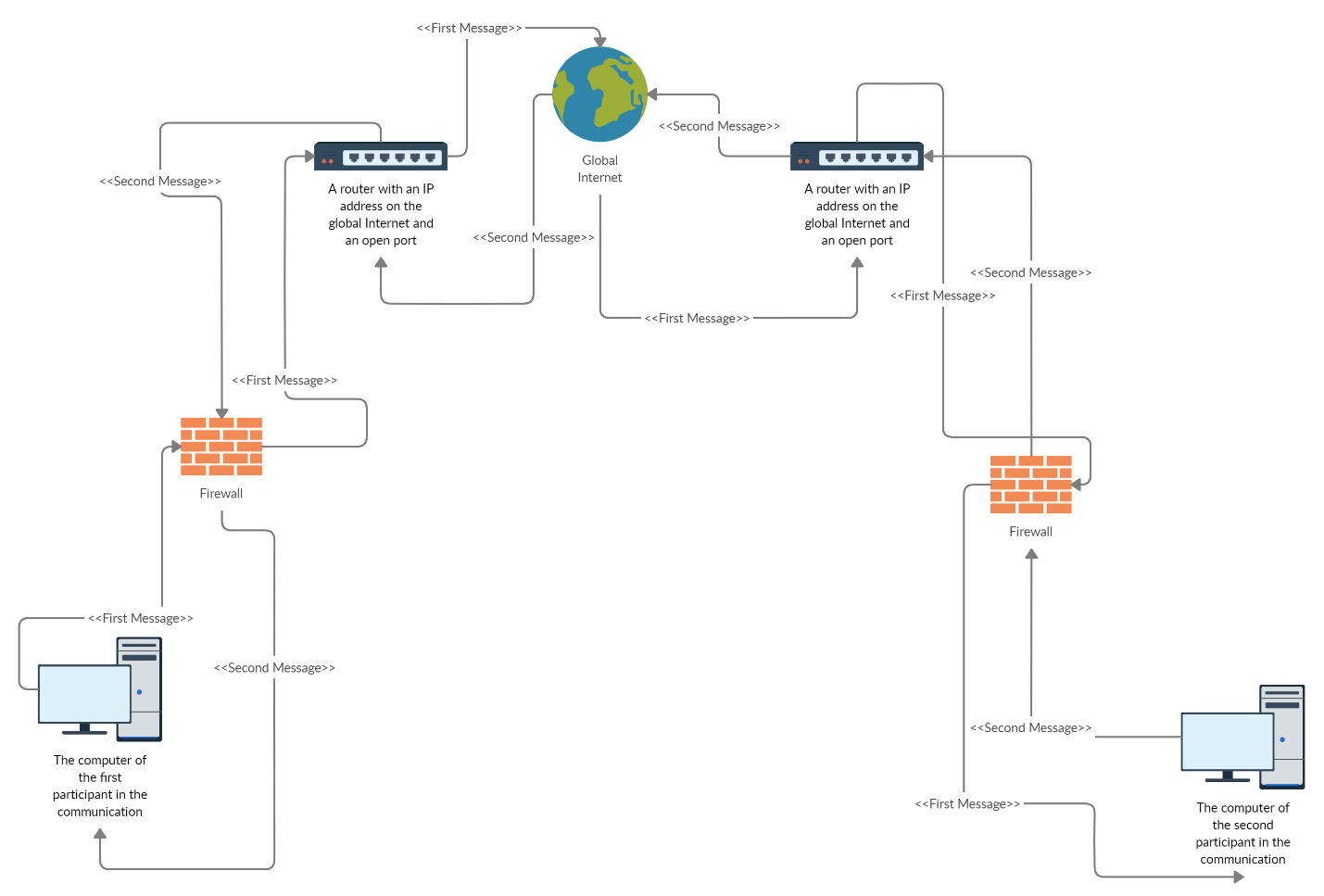
Приложение Е



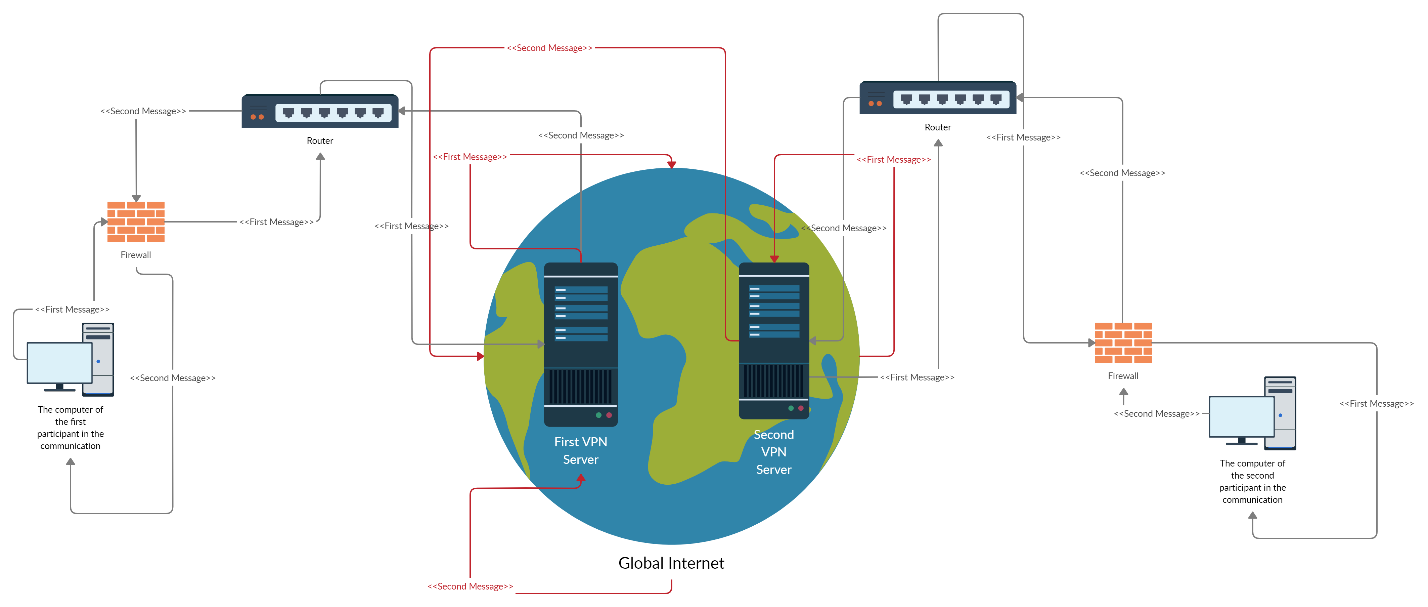
Приложение Ж



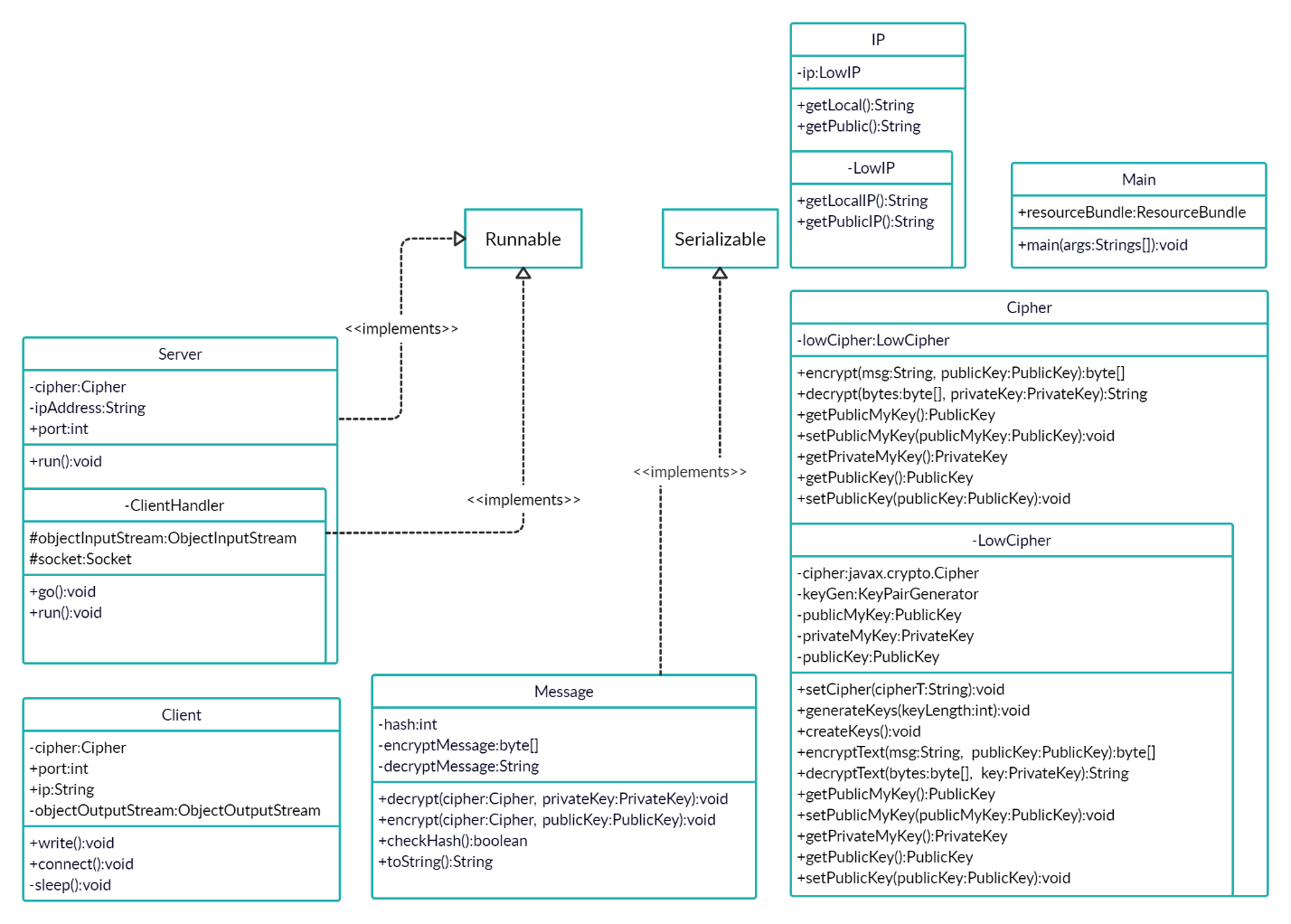
Приложение З



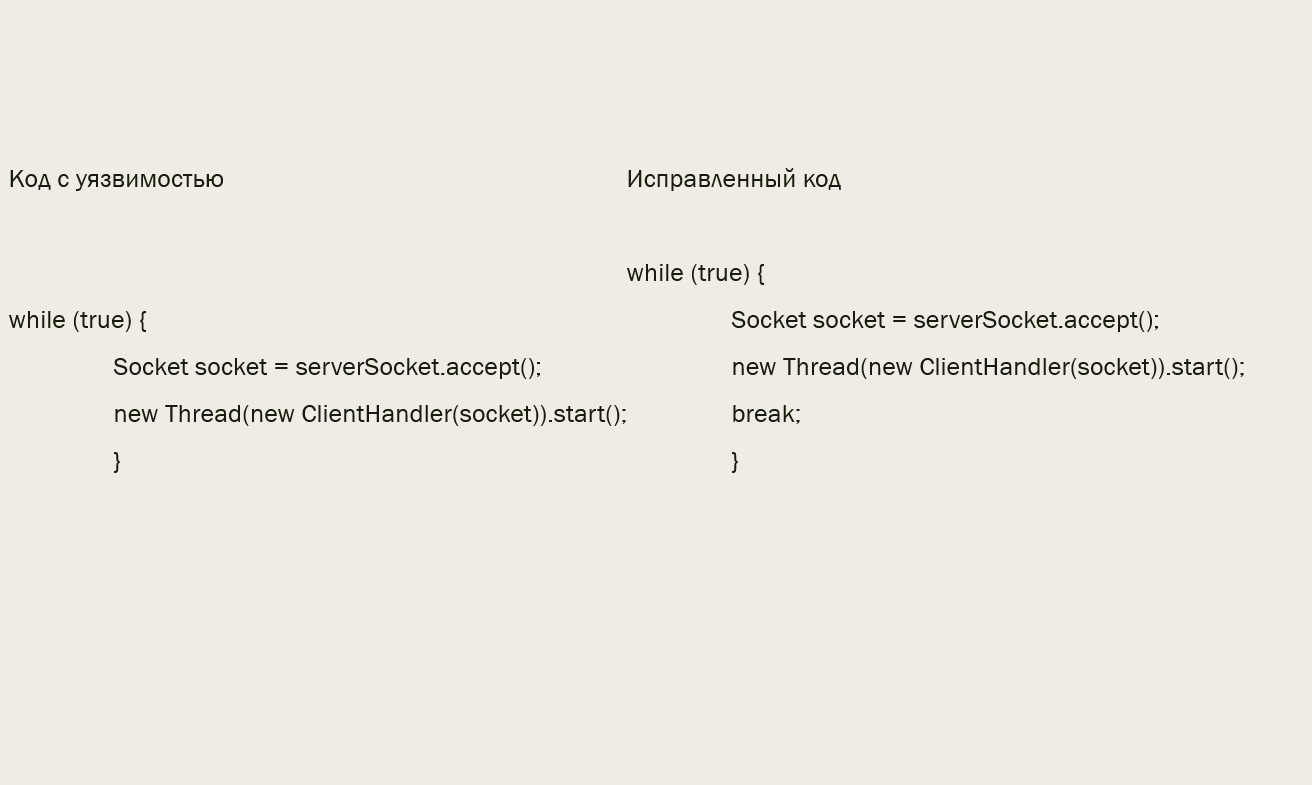
Приложение И



Приложение Й



Приложение К



Приложение Л



Приложение М

