МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО

«Пермский государственный национальный исследовательский университет»

**Отчёт**

по лабораторной работе №2 «Разработка распределенного приложения, использующего технологию передачи сообщений и сокеты»

по дисциплине «Технологии разработки распределенных приложений»

Работу выполнил Проверил

студент гр. ФИТ-8-2017 доцент кафедры

Габов Егор математического обеспечения

Владимирович вычислительных систем

23.12.2020 механико-математического факультета, к.ф.-м.н. доц. Деменев Алексей

Геннадьевич

Пермь, 2020 г

**Содержание**

[**Введение** 3](#_Toc57112407)

[**Основная часть** 4](#_Toc57112408)

[**Постановка задачи** 4](#_Toc57112409)

[**Выбор предметной области и инструментов** 4](#_Toc57112410)

[**Сценарии использования** 5](#_Toc57112411)

[**Описание приложения** 6](#_Toc57112412)

[**Заключение** 7](#_Toc57112413)

[**Литература** 8](#_Toc57112414)

# **Введение**

***Цель***: изучение возможностей технологии передачи сообщений и сокетов для создания распределенных приложений. Сравнение методов реализации взаимодействия компонент распределенной системы.

***Формируемые компетенции***: способность применять на практике теоретические основы и общие принципы разработки распределенных систем; способность использовать на практике стандарты сетевого взаимодействия компонент распределенной системы.

Разработать распределенное приложение, в котором:

1. Сервис обмена данными должен выполнять прием данных в нормализованную реляционную БД (например, спроектированную при выполнении входного контроля) из как минимум пять таблиц в 3-й нормальной форме.
2. Должно быть создано приложение, посылающее данные сервису при помощи сокетов и системы очередей сообщений, со свободной лицензией (Apache ActiveMQ, Apache Kafka или RabbitMQ), а при отсутствии такой возможности (соответствующих умений) допустимо использование импортных с бесплатной лицензией для университета в образовательных целях (например, MSMQ).
3. Данные перед передачей должны сжиматься и шифроваться при помощи ключа симметричного шифрования (DES).
4. Ключ симметричного шифрования должен передаваться сервису импорта для выполнения дешифрации данных.
5. При этом ключ симметричного шифрования должен в свою очередь шифроваться при помощи ключа асимметричного шифрования (RSA).
6. Ключ асимметричного шифрования должен генерироваться сервисом импорта и приложению должна передаваться открытая часть ключа.
7. Сервис импорта при получении данных должен импортировать их в БД при помощи механизма, реализованного при выполнении входного контроля.

# **Основная часть**

## **Постановка задачи**

Спроектировать, реализовать, протестировать и оценить приложение в соответствии с таблицей №1.

***Таблица 1. Распределение баллов за выполнение работы (***Максимальное количество баллов — 14)***:***

|  |  |
| --- | --- |
| **Требование к заданию** | **Максимальное количество баллов** |
| Приложение позволяет выполнять прием и передачу данных из ненормализованной БД в нормализованную без модификации данных. | 2 |
| Приложение позволяет передавать информацию с помощью очередей сообщений. | 3 |
| Приложение позволяет передавать информацию с помощью сокетов. | 3 |
| При передаче данных они шифруются с помощью симметричного ключа. | 2 |
| При передаче симметричного ключа шифрования данные шифруются с помощью асимметричного ключа. | 2 |
| Приложение, написанное студентом, работает в сети без сбоев. | 2 |

## **Выбор предметной области и инструментов**

Предметная область баз данных – БД банка. Главная сущность – кредиты.



**Рис. 1** Данные в ненормализованном виде

Ненормализованная таблица содержится в базе данных SQLite 3.33.0 [1,2]. Приложения для экспорта и импорта данных разрабатываются на языке программирования Python 3.8 [3,4]. В качестве IDE использовался PyCharm Community Edition [5]. PyCharm Community Edition является бесплатной версией, обладающей усеченным набором возможностей [6]. Таблицы в 3-й нормальной форме располагаются в базе данных MS SQL Server 2012 Express [7]. Для ассиметричного шифрования применяется модуль «RSA» в составе python-библиотеки «PyCryptodome» [8]. Для симметричного шифрования применяется модуль «DES» в составе python-библиотеки «PyCryptodome» [8].

## **Сценарии использования**

Приложение может использоваться в следующих сценариях.

1. **Передача данных с помощью сокета**

**Шаги выполнения:**

1. Заполнить файл imp\_config.ini, ввести ip-адрес и порт
2. Заполнить файл exp\_config.ini, ввести ip-адрес и порт
3. Пользователь1 запускает приложение передатчик
4. Пользователь2 запускает приложение получатель

**Ожидаемый результат:** Приложения обменяются ключами шифрования с уведомлением и выводом ключей на экран.

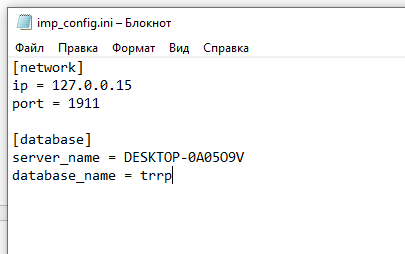
**Способ тестирования:**

При успешной передаче данных пользователям будет сообщено, что ключи получены, а сами ключи будут выведены в консоль.

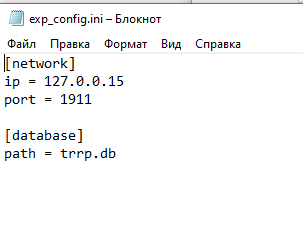
## **Описание приложения**

Приложение разделено на две части: передатчик (экспорт данных) и приёмник (импорт данных).

Приложение для импорта данных выступает в роли сервера, открывая сокет и ожидая подключения клиента. Приложение для экспорта данных выступает в роли клиента, подключаясь посредством сокета к серверу. Данные об ip-адресе и номере порта содержатся в файлах «imp\_config.ini» и «exp\_config.ini» для сервера и клиента соответственно.



**Рис. 2** Содержимое файла «imp\_config.ini»



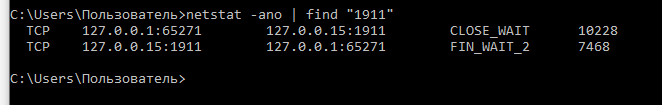
**Рис. 3** Содержимое файла «exp\_config.ini»

Чтобы убедиться в том, что программы используют для подключения сокеты, воспользуемся утилитой командной строки Windows «netstat» со следующими ключами:

• -a: Отображает все активные TCP-подключения и порты TCP и UDP, прослушиваемые компьютером

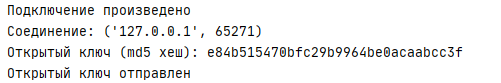
• -n: Отображает активные TCP-подключения, однако адреса и номера портов выражаются в числовом виде, и для определения имен не выполняется никаких попыток

• -o: Отображает активные TCP-подключения и включает идентификатор процесса (PID) для каждого подключения. Приложение, основанное на PID, можно найти на вкладке процессы в диспетчере задач Windows.



**Рис. 4** Вывод команды “netstat”

При установке соединения сервер генерирует пару открытый-закрытый ключ RSA и отправляет открытый ключ клиенту.

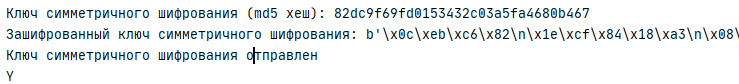


**Рис. 5** Генерация и отправка открытого ключа RSA сервером



**Рис. 6** Получение открытого ключа RSA клиентом

Затем клиент генерирует ключ шифрования DES, шифрует его открытым ключом RSA и посылает серверу. Сервер получает зашифрованный ключ и дешифрует его своим закрытым ключом

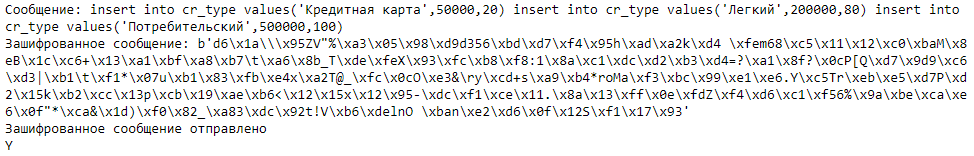


**Рис. 7** Генерация, шифрование и отправка ключа DES клиентом

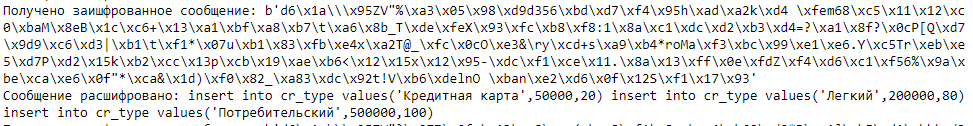


**Рис. 8** Получение и дешифровка ключа DES сервером

Далее данные пересылаются от клиента серверу в зашифрованном виде.

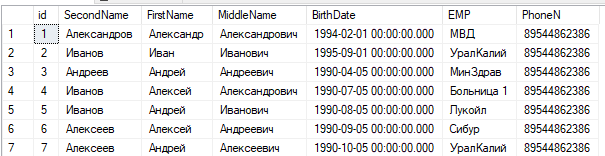


***Рис. 9.*** *Шифрование и отправка сообщения серверу клиентом*

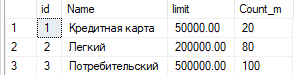


***Рис. 10.*** *Получение и расшифровка зашифрованного сообщения сервером*

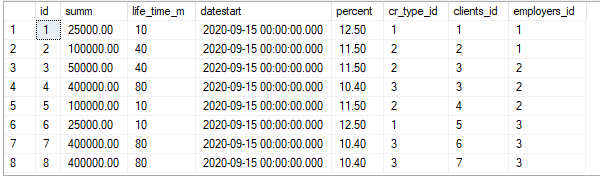
После окончания работы программ данные будут находиться в 5 таблицах в 3-й нормальной форме в базе данных MS SQL Server.



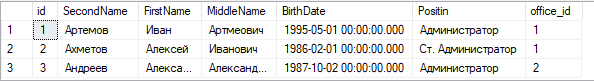
***Рис. 11.*** *Таблица «clients» (Клиенты) в MS SQL Server*



***Рис. 12.*** *Таблица «cr\_type» (Типы кредитов) в MS SQL Server*



***Рис. 13.*** *Таблица «credits» (Кредиты) в MS SQL Server*

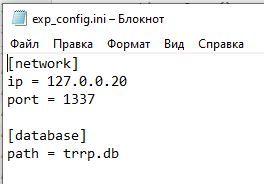


***Рис. 14.*** *Таблица «employers» (Сотрудники) в MS SQL Server*

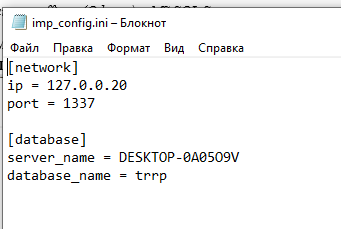


***Рис. 15.*** *Таблица « office» (Офисы) в MS SQL Server*

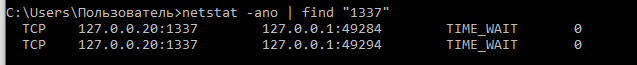
Запустим приложение с измененными параметрами сети. Наблюдаем, что работоспособность сохраняется.



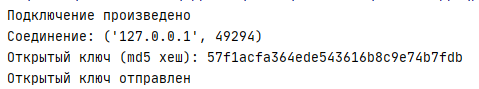
**Рис. 16** Измененное содержимое файла «exp\_config.ini»



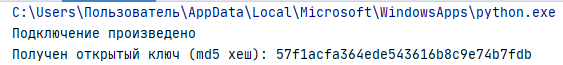
**Рис. 17** Измененное содержимое файла «imp\_config.ini»



**Рис. 18** Вывод команды “netstat”



***Рис. 19*** *Генерация и отправка открытого ключа RSA сервером*



**Рис. 20** Получение открытого ключа RSA клиентом

# **Заключение**

В ходе работы была достигнута цель и выполнены следующие требования:

* Приложение не позволяет выполнять прием и передачу данных из ненормализованной БД в нормализованную без модификации данных. 0 баллов.
* Приложение не позволяет передавать информацию с помощью очередей сообщений. 0 баллов
* Приложение позволяет передавать информацию с помощью сокетов. 3 балла
* При передаче данных они шифруются с помощью симметричного ключа. 2 балла
* При передаче симметричного ключа шифрования данные шифруются с помощью асимметричного ключа. 2 балла
* Приложение, написанное студентом, работает в сети без сбоев. 2 балла

Итого: 9 баллов

# **Литература**

* + 1. SQLite Download Page. url: [https://www.sqlite.org/download.html](https://www.sqlite.org/download.html%20) (Дата обращения 23.12.2020)
    2. SQLite Is Public Domain. url: [https://www.sqlite.org/copyright.html](https://www.sqlite.org/copyright.html%20) (Дата обращения 23.12.2020)
    3. Лицензия Python 3.8, <https://docs.python.org/3.8/license.html> (Дата обращения 23.12.2020)
    4. Python 3.8.0. Ссылка на скачивание URL: <https://www.python.org/downloads/release/python-380/> (дата обращения: 23.12.2020)
    5. PyCharm Community Edition. Распространяется под лицензией Apache 2, URL: <https://www.jetbrains.com/pycharm/features/editions_comparison_matrix.html> (Дата обращения 23.12.2020)
    6. PyCharm Community Edition. Ссылка на скачивание. URL: <https://www.jetbrains.com/pycharm/download/#section=windows> (Дата обращения 23.12.2020)
    7. Центр загрузки, URL: <http://www.microsoft.com/ru-ru/Download> (Дата обращения 23.12.2020)
    8. Библиотека PyCryptodome, URL: <https://pycryptodome.readthedocs.io/en/latest/src/license.html> (Дата обращения 23.12.2020)