МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Приложение для обмена сообщениями с использованием метода сквозного шифрования

Студенты: Сукочева Алис, Наместник Анастасия Андреевна

Группа: ИУ7-73Б

Руководитель: Рогозин Николай Олегович

МОСКВА, 2021 ГОД

Цель и задачи

Цель: реализовать метод сквозного шифрования для обмена сообщениями по не защищенному от прослушивания каналу связи

Для поставленной цели требуется решить следующие задачи

- 1. Проанализировать существующие решения
- 2. Описать метод решение поставленной задачи
- 3. Описать требования к системе
- 4. Разработать систему для решения поставленной задачи

Шифрование и дешифрование

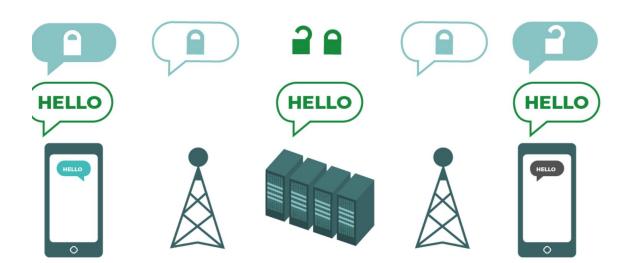
Шифрование - это процесс преобразования открытого текста в нечитаемый вид, чтобы защитить информацию, с применением математических алгоритмов и уникального набора бит, называемого ключом, к которому они применяется наряду с текстом

Дешифрование - это обратный шифрованию процесс с целью получить информацию в первоначальном виде



Способы шифрования данных при передаче

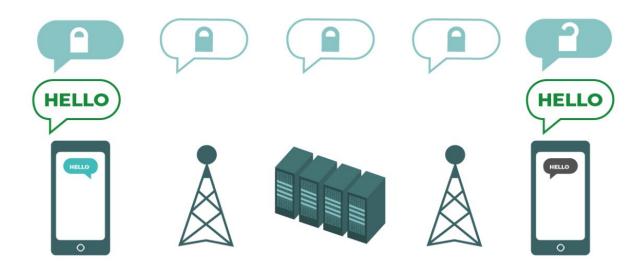
Шифрование транспортного уровня - предполагает наличие общего ключа отправителя и сервера, который используется для дешифрации сообщения на сервере и недоступен другим клиентам





Способы шифрования данных при передаче

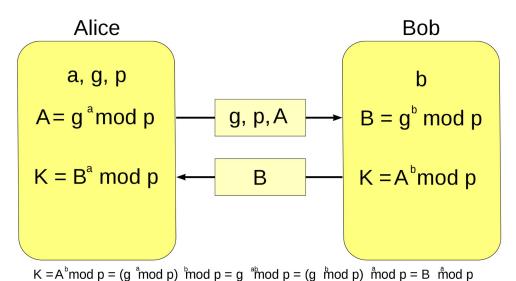
Сквозное шифрование - обеспечивает дешифрацию сообщения только участниками общения, что обеспечивает недоступность данных в исходном виде как для злоумышленника, перехватывающего пакеты, так и для промежуточного сервера



Протокол Диффи-Хеллмана

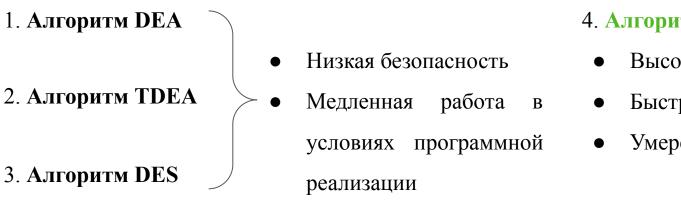
- Криптографический протокол, позволяющий двум и более сторонам получить общий секретный ключ, используя незащищенный от прослушивания канал связи.

Полученный ключ используется для шифрования дальнейшего обмена с помощью алгоритмов симметричного шифрования



Существующие решения

Симметричное шифрование - это способ шифрования, в котором для шифрования и дешифрования применяется один и тот же криптографический ключ



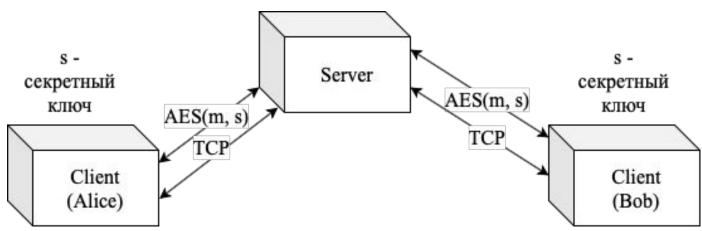
4. Алгоритм AES -

- Высокая безопасность
- Быстродействие
- Умеренные затраты памяти

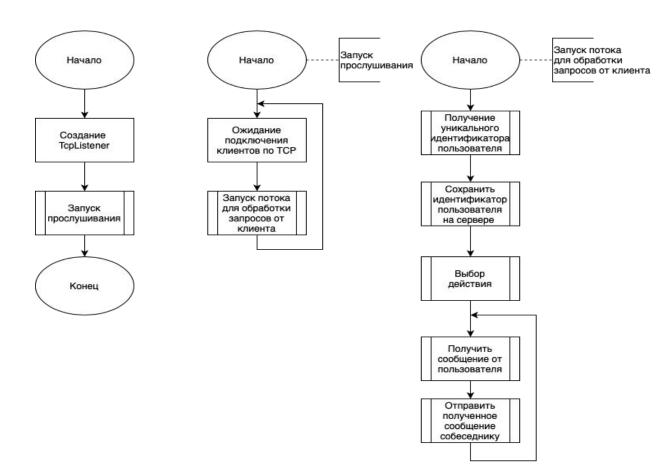
Проектирование системы

Система состоит из двух компонентов

- 1. Сервер
- 2. Клиент

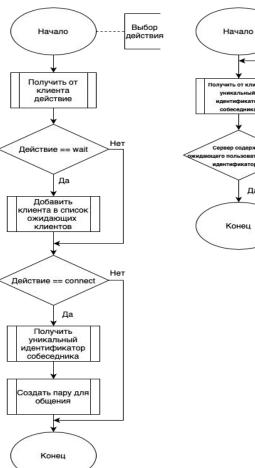


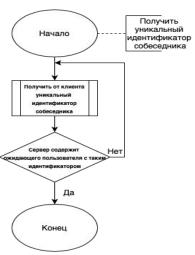
Схемы работы сервера



Схемы работы сервера







Схемы работы клиента

Начало

Подключение к

серверу

Получить

уникальный

идентификатор

Выбрать действие

Выполнить выбранное

действие

Запустить

поток для

отправления сообщений

Запустить

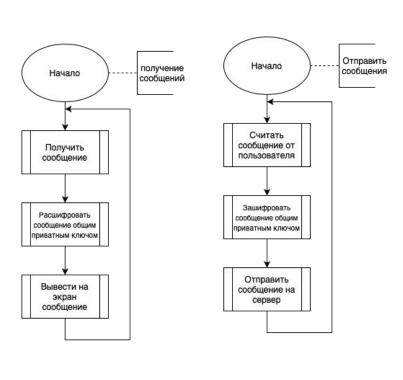
поток для

получения

сообщений

Отправить сообщения

Конец



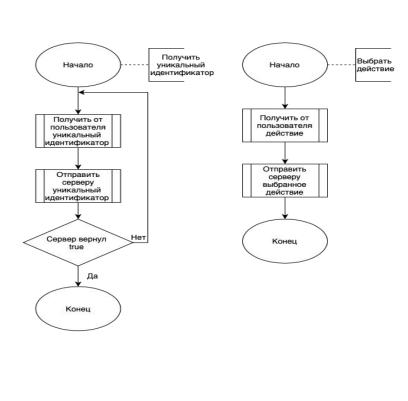
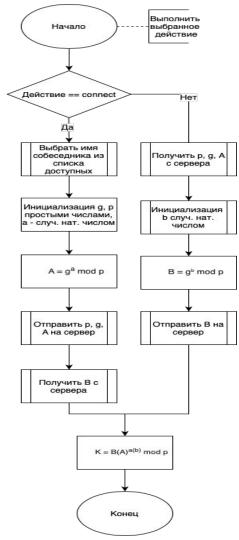
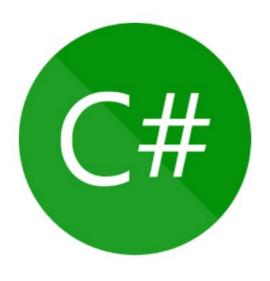


Схема протокола Диффи-Хеллмана на стороне клиента

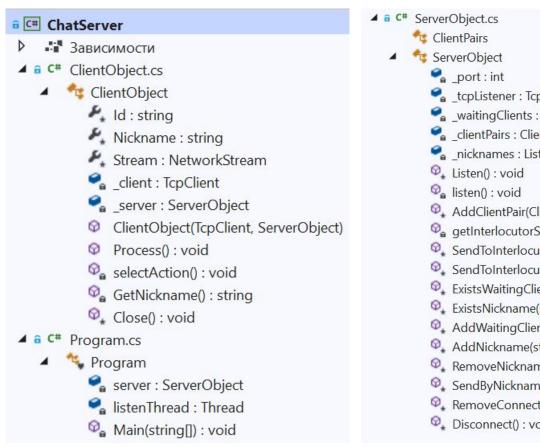


Технические средства реализации



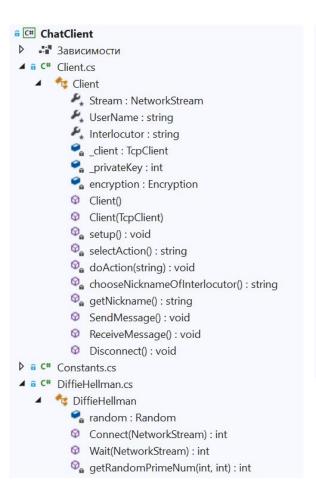


Структура и состав классов (Сервер)



```
tcpListener : TcpListener
waitingClients : List<ClientObject>
clientPairs : ClientPairs
anicknames : List<string>
AddClientPair(ClientObject, string): void
getInterlocutorStream(string) : NetworkStream
SendToInterlocutor(string, string): void
SendToInterlocutor(string, byte[]): void
ExistsWaitingClient(string): bool
ExistsNickname(string): bool
AddWaitingClient(ClientObject): void
AddNickname(string): void
RemoveNicknames(string): void
SendByNickname(string, string): void
RemoveConnection(string): void
Disconnect(): void
```

Структура и состав классов (Клиент)



```
▲ 6 C# Encryption.cs

       Encryption
          key : byte[]
             Encryption(byte[])
          ToAes256(string) : byte[]
            FromAes256(byte[]): string
             CreatyKeyByInt(int, [int]): byte[]

▲ 6 C# MyMath.cs

       MyMath
             IsNumberPrime(int): bool

▲ 6 C# Program.cs

       Program
          client : Client
          Main(string[]): void
```

Пример работы системы

```
Enter your name: Bob
Select action: wait; connect: connect
Enter the name of the interlocutor: Alice
privateKey (connect) = 12044
[You communicate under: Bob]
Enter the message:
Hi, Alice
Alice: Hi, Bob
```

```
Первый клиент (Bob)
```

```
Enter your name: Alice
Select action: wait; connect: wait
privateKey (wait)= 12044
[You communicate under: Alice]
Enter the message:
: Hi, Alice
Hi, Bob
```

Второй клиент (Alice)