Правительство Севастополя

Департамент образования

ГБОУ «ЦДОД «Малая академия наук города Севастополя»

Отделение: компьютерные науки

Секция: Безопасность информационных и   
телекоммуникационных систем,   
технологии программирования

**Шифрование сообщений: CryptoChat v.1.0**

**Работу выполнил:**

Цапля Дмитрий Григорьевич,

ученик 9 класса Гимназии №1

Верещака Алексей Николаевич,

ученик 9 класса Гимназии №1

**Научный руководитель:**

Глеч Екатерина Викторовна,

педагог ДО ГБОУ “ЦДОД “Малая академия наук города Севастополя”

Севастополь – 2017

**СОДЕРЖАНИЕ**

Введение.................................................................................

1.Теоретический раздел.........................................................

2.Практическая реализация.............................................

3.Приложение на базе Android...............................................

Заключение............................................................................

Библиографический список..................................................

**Введение**

В наше время как никогда важен обмен информацией между людьми. Как и любая другая система обмена информацией, Интернет ненадежен. Сейчас информация - самое ценное, что есть у людей. Не зря современное общество называют информационным. И многие хотят получить доступ к чужой информации.

Во все времена люди пытались решить проблему защиты информации. Человечество придумало шифрование, защищенные пути передачи, но эти способы все еще не являются абсолютно надежными.

За всю историю существования человечества было придумано более десяти тысяч основных методов шифрования, количество которых можно увеличить до сотен миллионов. Шифры использовали еще во времена античности - например, известнейший шифр Цезаря. В настоящее время, как шифрование, так и дешифрация производится с помощью мощных суперкомпьютеров. И задача сегодняшних шифровальщиков - защитить информацию пользователей так, чтобы ее было невозможно расшифровать, даже если она будет перехвачена. Современные мессенджеры все чаще и чаще подвергаются кибератакам. Фактически, современные хакеры способны взломать почти любой сервер, имеющий доступ к интернету. Поэтому защитить информацию пользователей на 100% невозможно - даже самая надежная сеть имеет уязвимости.

Эта проблема заинтересовала нас, и было решено начать изучение проблемы шифрования с азов - шифра Цезаря. У него слишком слабая криптостойкость, поэтому мы решили использовать шифр Вижинера, криптостойкость которого при десятизначном ключе очень высока. А при ключе, длина которого равна длине сообщения, взлом практически неосуществим.

Следующая проблема, которую мы решали, была передача ключа к шифру от одного пользователя к другому. Решено было использовать криптографический протокол Диффи -Хеллмана, с помощью которого можно создать одинаковый ключ у бесконечного количества пользователей, не передавая его от одного пользователя к другому. Таким образом, предавая информацию и шифруя ее методом Вижинера по ключу, созданному по протоколу Диффи - Хеллмана, можно говорить об очень высокой криптостойкости. Этот протокол представляет огромные возможности для создания уникальных ключей, ведь работая с большими числами(2^1000 и больше) можно получить огромный одинаковый ключ длиной в 100 букв и больше.

Мы предлагаем пользователям удобный чат, который обеспечивает безопасность и конфиденциальность передаваемых данных. В настоящее время разработана версия для стационарных компьютеров, ведется разработка мобильного приложения для Android.

1. **Теоретический раздел.**

Шифр — какая-либо система преобразования текста с секретом (ключом) для обеспечения секретности передаваемой информации. Шифр Виженера (фр. Vigenere) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Шифр Виженера изобретался многократно. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа. Данный шифр основан на еще более старом шифре - шифре Цезаря. В шифре Цезаря каждая буква алфавита сдвигается на несколько позиций:

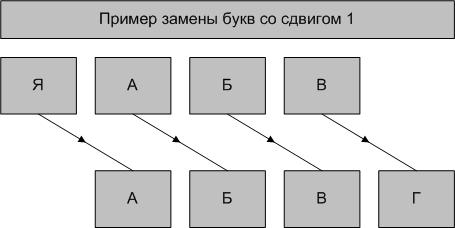


Рис. 1.1 – Шифр Цезаря с ключом 1

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

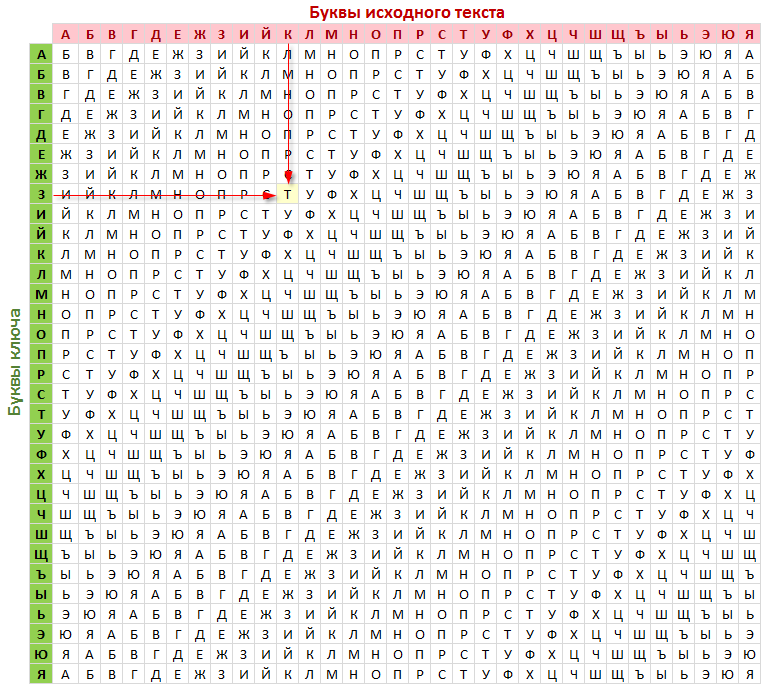


Рис. 1.2 – «Квадрат» Виженера

Для программной реализации шифра Виженера используется таблица символов ASCII - cтандартный набор из 256 символов, включающий в себя большие и маленькие русские и латинские буквы, цифры, знаки препинания и знаки арифметических действия и т. п. Каждому символу ASCII соответствует 8-битовый двоичный код.

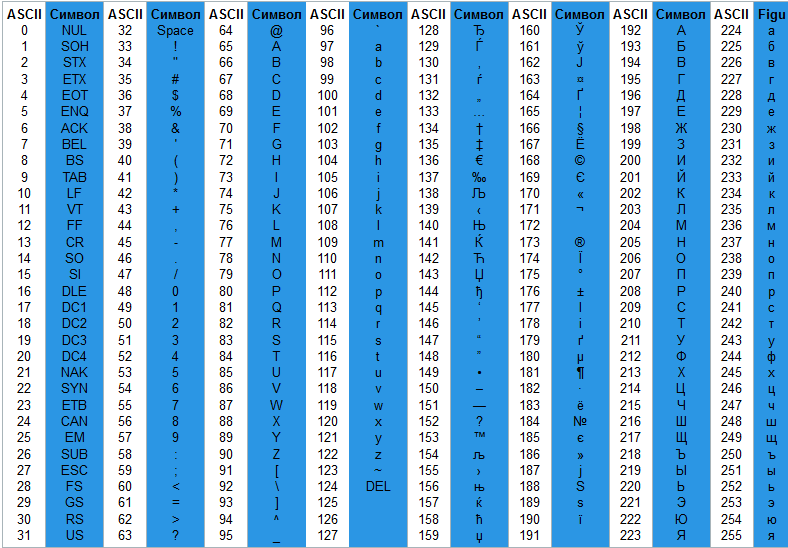


Рис. 1.4 – Таблица символов ASCII

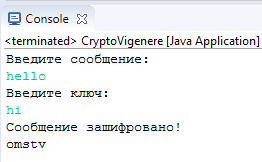


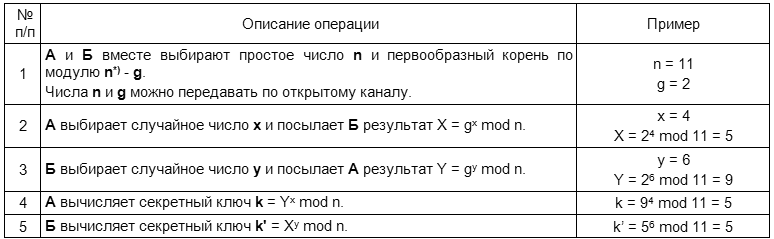
Рис. 1.5 – Пример выполнения консольной программы

Протокол Дифффи - Хелллмана (англ. Diffie-Hellman) — 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.

Схема открытого распределения ключей, предложенная Диффи и Хеллманом, произвела настоящую революцию в мире шифрования, так как снимала основную проблему классической криптографии — проблему распределения ключей. Их вкладом в криптографию было убеждение, что ключи можно использовать парами — ключ зашифрования и ключ расшифрования — при условии, что исключается возможность определения содержимого ключа для расшифрования исходя из содержимого открыто передаваемого ключа для зашифрования.

В чистом виде алгоритм Диффи-Хеллмана уязвим для модификации данных в канале связи, в том числе для атаки «Человек посередине», поэтому схемы с его использованием применяют дополнительные методы односторонней или двусторонней аутентификации.

Таблица 1.1 – Обмен ключами по протоколу Диффи-Хеллмана



1. **Описание разработки**

Приложение реализовано на языке программирования Java.

Для работы программы необходима установленная jre8:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

Инсталляция и запуск:

1. Запуск cryptochat.jar (исполняемый файл) двойным щелчком мыши;
2. Запустить сервер Create new server;
3. Запустить клиент Connect.

Серверная часть отвечает лишь за передачу зашифрованных сообщений и соединение пользователей. Клиентская часть совершает шифрование и дешифрование сообщений и отвечает за графический интерфейс.

Было решено использовать модель мессенджера на сокетах для обмена информацией через классы InputReader и OutputWriter.

Описание работы мессенджера.

Наша программа состоит из 4-х основных частей: клиента, персонального сервера, глобального сервера и лаунчера.

**1)Лаунчер**

При запуске программы клиент видит лаунчер - окно, в котором есть 2 кнопки и текстовое поле. Первая кнопка позволяет создать свой персональный сервер (разговор, конференцию). Вторая кнопка позволяет присоединиться к уже созданному серверу, обратившись к нему по его ip - адресу. Список активных серверов отображается в текстовом окне.

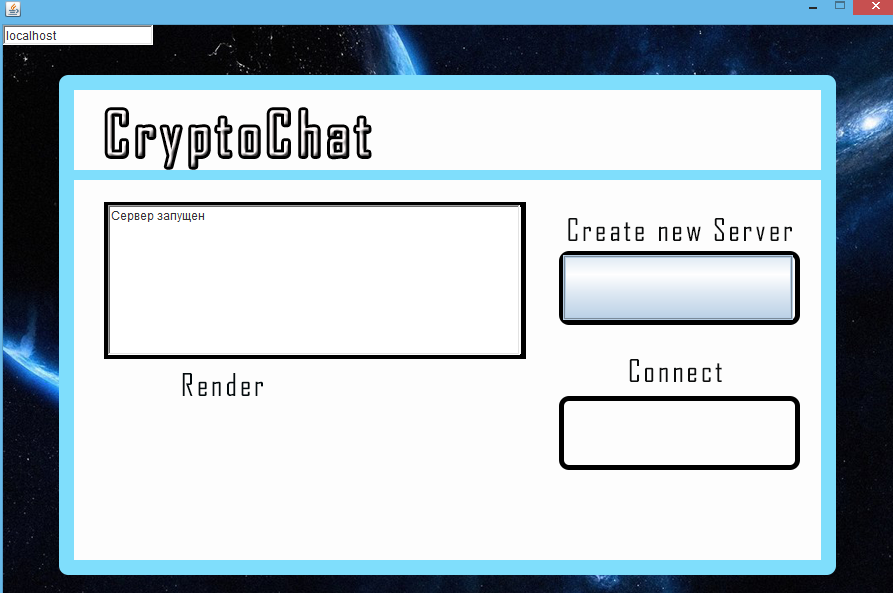


Рис. 2.1 – Окно лаунчера

**2)Клиентская часть**

Присоединившись к доступному серверу, пользователь подключается к чату. В клиентской части происходит обработка действий пользователя, ввод и вывод данных, отправка сообщений на сервер, и, конечно же, создание ключа и шифрование. Когда на сервере появляется больше чем 1 пользователь, клиент совместно с локальным сервером создает ключ для шифра по протоколу Диффи - Хеллмана длиной в 10 символов. Длина среднего сообщения в социальных сетях не более 20 символов. Ключ такой длины позволяет избежать взлома шифра с помощью частотного анализа - процесса, при котором используются сведения о частоте использования тех, или иных букв в языке.

Далее пользователю доступно поле для ввода сообщений, кнопка для их отправки, лог(текстовое поле), где отображаются сообщения всех клиентов, подключенных к серверу, а также же лог, где можно будет с помощью нажатия на кнопку дешифровать лог сообщений. Таким образом, мы отделили процесс шифрования данных от сети. Стороннее лицо не сможет восстановить сообщение, перехватив его. Все истории сообщений хранятся в файле на ПК пользователей. Таким образом, мы предотвращаем атаки на глобальный сервер.

Наша система полностью анонимна и не требует регистрации, т.к. для идентификации пользователей мы используем их ip - адреса. При этом создается проблема, что доступ к истории своих сообщений можно получить со своего домашнего адреса. Она может быть решена с помощью возможности добавления к одному главному ip дочерних ip, с которых будет открыт доступ. Эта часть пока не реализована.

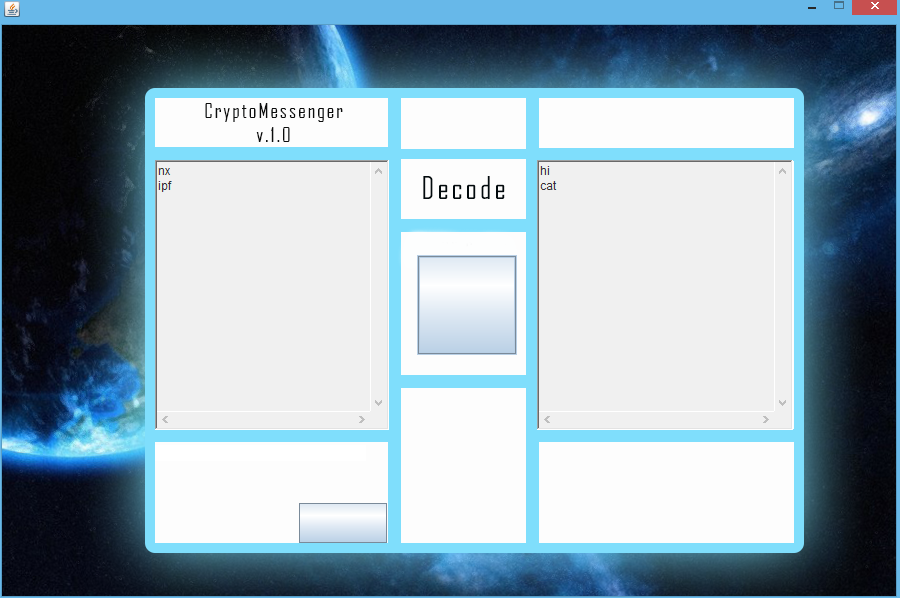


Рис. 2.2 – Клиентская часть

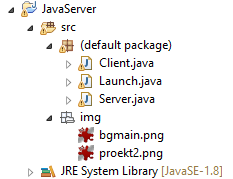
**3)Серверная часть**

Эта часть программы запускается из лаунчера, но пока не имеет графического интерфейса. Пользователь сможет лишь увидеть свой сервер в списке и при желании удалить его. Сам сервер при подключении к нему клиентов генерирует ключ совместно с ними и производит обмен сообщениями между ip адресами. Персональный сервер - лишь средство для передачи уже зашифрованной информации.

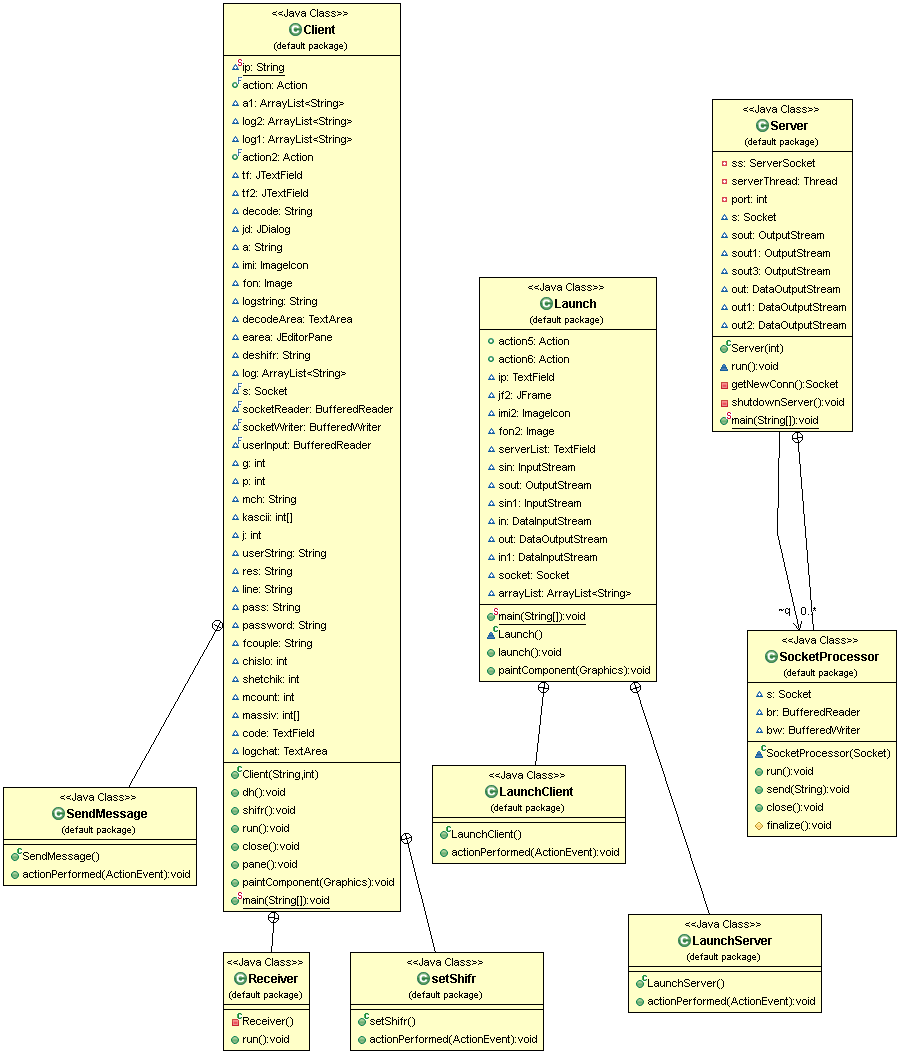
**4)Глобальный сервер**

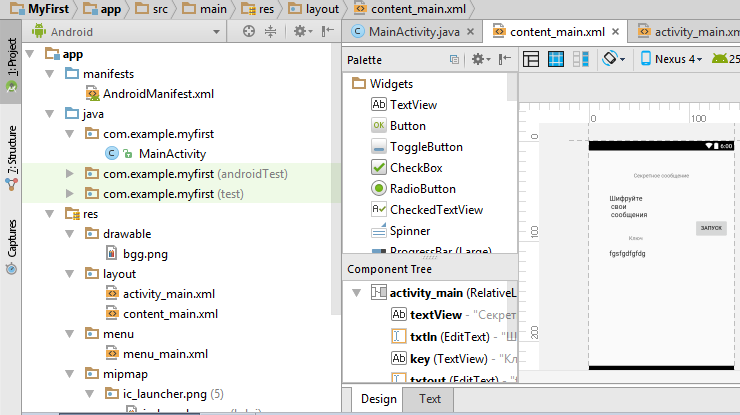
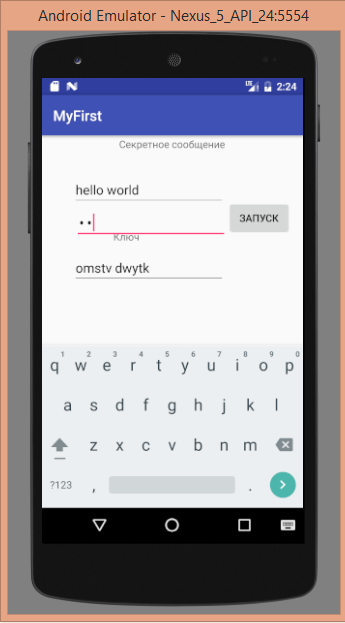
Эта часть программы пока не реализована. Она служит для хранения списка активных серверов, то есть является своеобразной базой данных. Она обменивается данными с лаунчером в отдельном потоке программы и никак не влияет на работу клиента и персонального сервера.

Структура проекта в IDE Eclise:



UML-диаграмма приложения:





**Заключение**

В нашей работе было проведено исследование в области шифрования и передачи информации и было предложено решение проблемы ее безопасной передачи с помощью создания мессенджера с шифрованием при передаче информации. Мы активно занимаемся разработкой нашего проекта и в наших планах - создание глобального сервера, модификация программы и создание полноценного кроссплатформенного приложения. Сейчас мы занимаемся изучением разработки приложений на базе Android и уже создали мобильное приложение, которое шифрует сообщения методом Вижинера.

Этот проект дал нам огромный опыт, как в разработке работающего приложения, так и в сфере проектирования.

**Список использованных источников**

1. Баричев С. В. Криптография без секретов. –М.: Наука, 1998. –120 с.
2. Сухов С. А. Учебное пособие Основы программирования на Java. Ульяновск: УлГТУ, 2006. - 88 с.
3. Харди Б., Филлипс Б. Android. Программирование для профессионалов. 2-е изд. – СПб.: Питер, 2016. – 640 с.
4. Эккель Б. Философия Java. 4-е изд. СПб.: Питер, 2009. – 637с.

Интернет-источники:

1. <https://cryptoworld.su/>
2. <https://developer.android.com/studio/index.html>
3. <http://www.android-ide.com/>