1: Титульный слайд:

Здравствуйте, меня зовут Малышев Тимур. Я обучающийся МАОУ «Школа № 74 с УИОП» Московского района города Нижнего Новгорода.
Тема моей работы: Разработка приложения «Cryptographer — Шифровальщик» для смартфона под управлением ОС Android
2 слайд:

Сегодня все больше людей общаются не лично, а через компьютерную сеть. Ничто не ценится так дорого, как информация, записанная в цифровой форме и передаваемая по сетям. И ничто другое невозможно так легко потерять, как эту информацию. Для обеспечения сохранности информации при передачи сообщения я создал свое приложение «Шифровальщик».

3: Цель и задачи проекта:

Целью данной работы является разработка приложения с графическим интерфейсом для смартфона под ОС Android для анонимного общения в сети.
Для достижения поставленной цели предполагается решение ряда задач:

- Изучение алгоритмов шифрования;
- Анализ существующих приложений;
- Проектирование структуры приложения;
- Программная реализация;
- Тестирование полученного продукта;
- Организация поддержки программного продукта.

4: Инструменты разработки:

Под мобильную ОС Android я разрабатывал и тестировал свой проект, используя следующие инструменты:
Gradle - система сборки проекта
Java - язык программирования
Kotlin - потомок языка Java
Intellij Idea – IDE (интегрированная среда разработки), в которой я создаю все проекты, как консольные так и под Android.

5: Структура приложения:

Приложение работает с Android 4.2(Jelly Bean) API LEVEL=17 и до последней, на сегодняшний день, Android 10-11, API=29-30

но из-за некоторых зависимостей шифры BlowFish и AES работают начиная с Android 8.0(Oreo), API LEVEL=26.

Чтобы удобнее было писать новые классы, я был создан главный класс - StartActivity.

Он несет функцию объединения всех остальных классов.

Этот класс содержит в себе функции всех кнопок, переходов и так далее.

А остальные классы содержат только структуру алгоритма или шифра.

Именно таким подходом достигается высокая скорость разработки приложения.

Немного про фрагменты.

Android добавил фрагменты с API 11 для разработки более гибкого пользовательского интерфейса.

(использовались при создании главное меню приложения).

Т.к. фрагменты не могут существовать без классов, то был создан главный класс - MainActivity.

Класс MainActivity - является первым классом, который видит пользователь.

StartActivity - объединяет классы, а MainActivity - отображает их,

т.е. главная функция этого класса - это создание пользовательского интерфейса.

6-7 слайды:

В ходе работы над проектом были изучены следующие алгоритмы: DES, AES, BlowFish, RSA. Так же изучены шифры подстановки. В процессе работы был разработал алгоритм шифрования, основанный на шифре Цезаря, который получил название eXT.

Принцип работы приложения с выбором алгоритма eXT:

Вы вводите своё сообщение, выбираете метод (четный/нечетный), по умолчанию стоит четный.

Затем вы вводите пароль. Пароль может состоять только из чисел.

Слова и другие символы вы ввести не сможете т.к. открывается клавиатура только с числовыми значениями.

Если вы не введете пароль, то всё равно получите 2 ключа потому, что я создал пароль по умолчанию, который знаю только я.

Далее из пароля извлекаются два ключа, после выбора ключа, нажав на кнопку «Далее», вы получаете своё зашифрованное сообщение.

8 слайд

Алгоритм eXT

Из пароля, который должен быть больше 999, извлекаются цифры.

Если пользователь не ввёл пароль, то ставится пароль по умолчанию.

Например: пароль=1234, извлекаются 1 2 3 и 4. (a=1, b=2, c=3, e=4)

Далее эти числа представляют квадратное уравнение:

ax^2+bx+c=e,

далее уравнение преобразуется: ax^2+bx+c-e=0 -> ax^2+bx-1=0.

Потом вычисляется дискриминант: Дискриминан = b * b - 4*a*c.

Потом вычисляются иксы(x):

x1 = (-b + Дискр) / (2*a)

x2 = (-b - Дискр) / (2*a)

x1 и x2 - это и есть наши ключи.

Далее каждая буква сообщения меняется.

Если пользователь выбрал нечетный метод, тогда:

Извлекается код буквы, и к нему прибавляется x1.

Например, буква А имеет код=1040, x1=3(к примеру) =>

код=1040+3 а это буква "Г".

Если метод четный, тогда из кода буквы вычетается x1=3 =>

"А" превратиться в "Э".

Таким образом шифруется/дешифруется всё сообщение.

9 -10 слайды

В работе также реализован алгоритм шифрования криптографический алгоритм с открытым ключом RSA,

основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Как пользоваться?

* Создать ПУБЛИЧНЫЙ ключ.

* Дать ключ другу.

* Друг зашифрует сообщение публичным ключом и передаст вам зашифрованное сообщение.

* Используя ПРИВАТНЫЙ ключ, вы расшифруете зашифрованное сообщение.

* Главное: публичный ключ можно дать кому угодно, а приватный должен быть только у вас!

11 слайд выводы

За время работы над приложением:

- были изучены различные алгоритмы шифрования (AES, DES, BlowFish, RSA);
- изучены шифры подстановки;
- приобретены навыки программирования на языках Java и Kotlin;
- изучен язык разметки XML.

Благодаря этим знаниям было создано приложение для смартфона под управлением ОС Android «Cryptographer — Шифровальщик». Продукт прошел тестирование среди учащихся школы и получил хорошие отзывы. В будущем планируется загрузка приложения в Google Play и организация поддержки программного продукта.

Про другие приложения:

Минусы программы "Генератор текста":

*) У этого приложения нет меню

*) Оно имеет в своем наборе только алгоритм Цезаря

Платные программы

В моём приложении:

*) Есть меню

*) Есть алгоритм Цезаря и еще другие методы шифрования.

2) Минусы программы "Шифроватор текста":

*) Приложение имеет интерфейс, который разработчики приложений для андроид используют по умолчанию.

*) В нём нельзя именять цвета элементов.

*) Приложение имеет только 1 способ шифрования текста

В моём приложении:

*) Можно изменить цвет всех элементов интерфейса.

*) Как я и говорил ранее: Помимо одного способа шифрования текста, у меня их несколько.

Далее всё на слайдах...