САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

РАЗРАБОТКА СИСТЕМЫ ВЫБОРОЧНОГО ШИФРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ

Студент группы 6511-100503D
Коган Артур Эдуардович

Самара 2019

Цель: разработка инструмента шифрования фрагмента текста, выбираемого пользователем.

Задачи:

- Анализ существующих сторонних решений по шифрованию передаваемых пользовательских данных.
- Разработка архитектуры и пользовательского интерфейса приложения.
- Реализация приложения выборочного шифрования пользовательских данных.
- Проверка функциональных возможностей и корректности работы приложения.

В современном мире для передачи данных в зашифрованном виде могут использоваться как готовые программные решения, так и разработанные пользователем для себя:

– E-Mail.
  GMail, Mail.Ru, Yandex.Mail, Outlook.Com, ProtonMail, iCloud Mail, …
– **Социальные сети**.
  VK, Facebook, LinkedIn, Twitter, Instagram, Tumblr, …
– **Мессенджеры**.
  WhatsApp, Viber, Telegram, ICQ, qip, …
– **Клиенты и веб-версии различных чатов**.
  Skype, Telegram, Viber, …

Наиболее развитые приложения

1) ProtonMail

Особенности:

–Шифрование, основанное на открытом исходном коде.
–Нулевой доступ к данным пользователя.
–Сквозное шифрование.

Недостатки:

–Необходима регистрация для начала использования.

2) Delta Chat

Особенности:

–Лицензия GPLv3.
–Autocrypt.
–Push-IMAP уведомления.

Недостатки:

–Использование E-Mail серверов.

3) PGP

Особенности:

–Использование ключевой пары.
–Цифровая подпись.
–Хеширование.
–Сжатие данных.

Недостатки:

–Проблема хранения долговременных ключей.

4) ZeroTier One

Особенности:

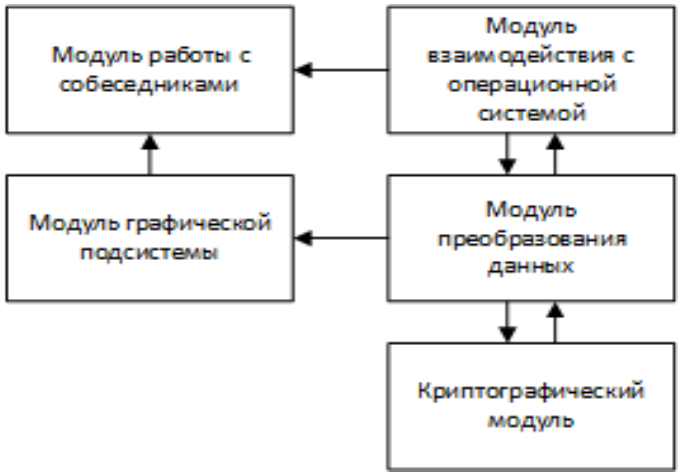
–Открытый исходный код.
–P2PVPN.

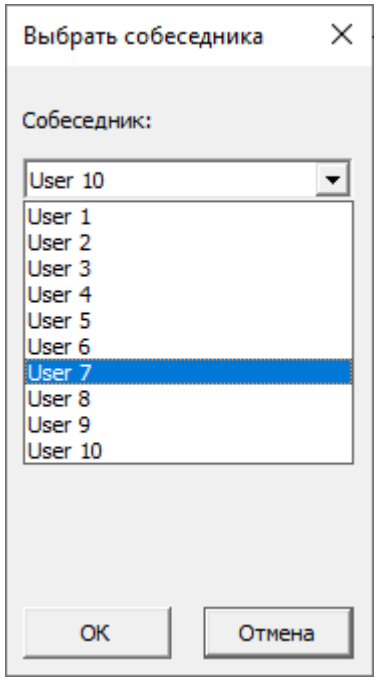
Недостатки:

–Не шифрует данные, посылаемые на сторонние сервера, как любая VPN.

Возможности:
– Шифрование / дешифрование.
– Горячие клавиши.
– Управление собеседниками.
– «Контактная книга», многопользовательский режим.

Краткое описание работы приложения:
1) Обработка нажатия горячей клавиши.
2) Предобработка данных.
3) Шифрование/дешифрование.
4) Постобработка данных.





Алгоритм шифрования: AES. Является стандартом, хорошо изучен и распространен повсеместно.

Ключ шифрования: сеансовый. Наличие большого объема данных, зашифрованных одним и тем же ключом, дает возможность подобрать этот ключ шифрования, например статистическим методом.

Генерация сеансового ключа: Argon2. Является победителем конкурса Password Hashing Competition. Функция Argon2 способна контролировать затрачиваемое на хеширование время и память и делает атаки полного перебора бессмысленными. Данная функция распространена повсеместно, и имеются реализации с открытым исходным кодом. Для генерации сеансового ключа используется долговременный ключ и «одноразовое» случайное значение Nonce. Nonce передается вместе с зашифрованным сообщением.

Хранение долговременного ключа: зашифрованный файл. При запуске приложения запрашиваем пароль пользователя и дешифруем им файл. В файле хранятся долговременные ключи каждого из собеседников.

Универсальность: Для корректной передачи и отображения в любом приложении, зашифрованный текст кодируется с помощью Base64.

Описание процедур шифрования и дешифрования

Алгоритм шифрования

Шифрование данных

Используемый алгоритм шифрования: AES-128

$$SK = Argon2(key, N_{once})$$

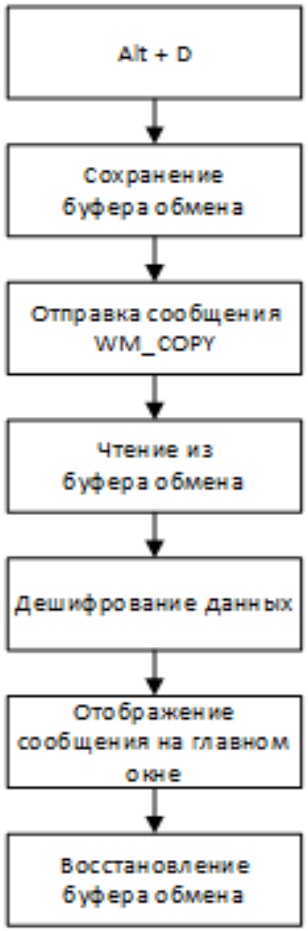
$$C = (N_{once} \,||\, E_{SK}(P))$$

$$P = D_{SK}(C)$$

Пример отправляемых данных:

nVKhmrLP	rmjSmd8Wv2YnOKZHo3M0fw=
Nonce	Base64 шифротекст

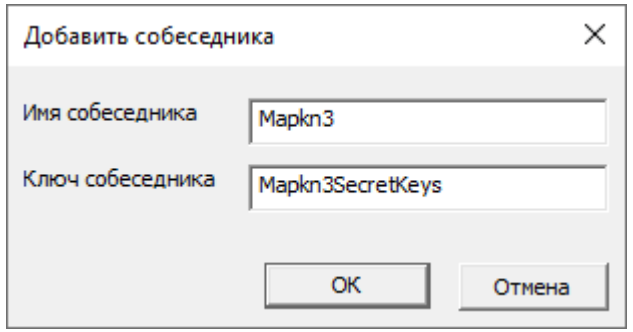
Алгоритм дешифрования

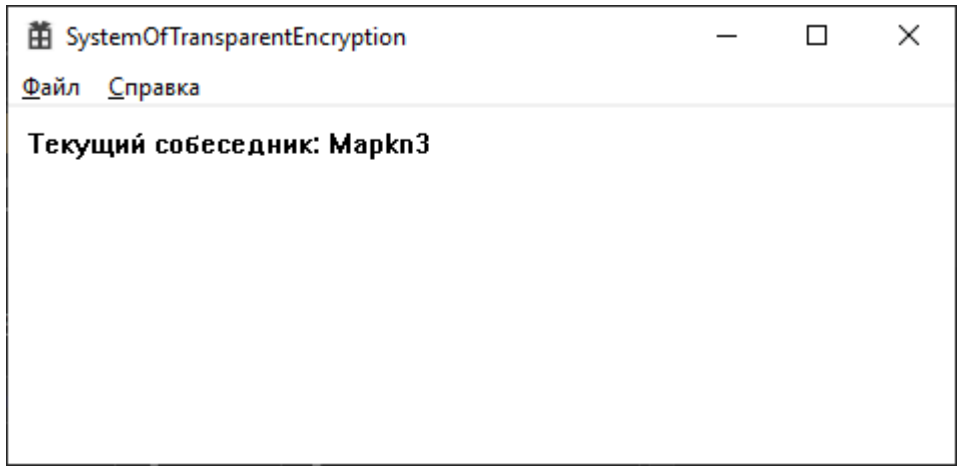


Дешифрование данных

Для добавления в приложение нового собеседника нужно сделать следующее:

1) Нажать Alt+N для вызова формы добавления нового собеседника.
2) Заполнить поля имени и ключа собеседника.
3) Нажать OK.
4) Проверить, что на главном окне приложения указан верный текущий собеседник.

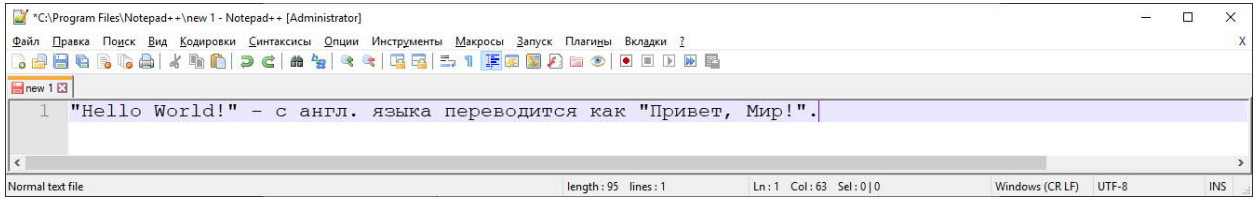


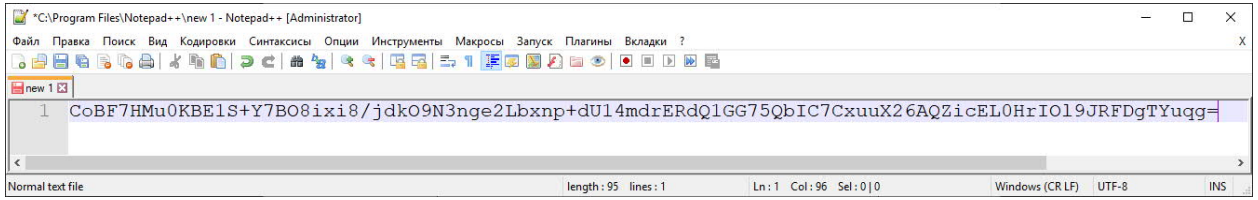


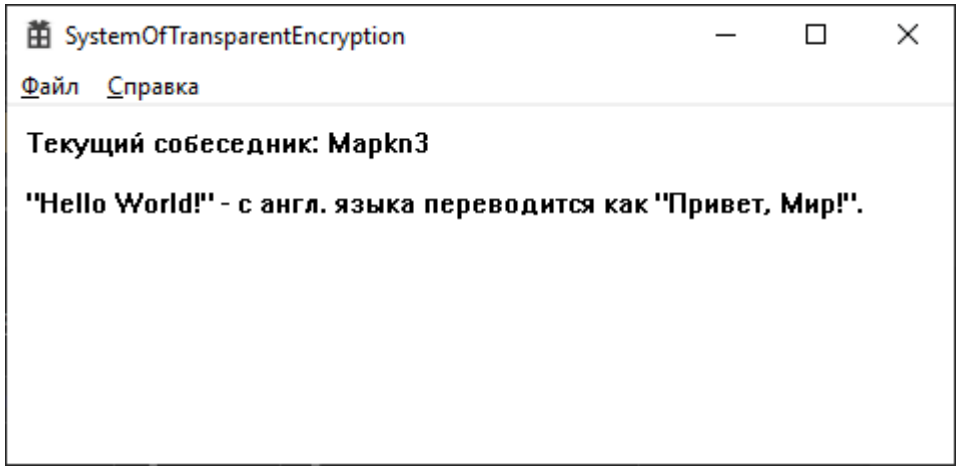
Проверка работоспособности и корректности работы разработанного приложения

Для проверки корректной работы шифрования и дешифрования выполним следующие шаги:

1) Открыть текстовый редактор.
2) Набрать любое проверочное сообщение.
3) Выделить сообщение.
4) Нажать Alt+E для шифрования сообщения.
5) Выделить полученный шифротекст.
6) Нажать Alt+D для дешифрования выделенного текста.
7) Проверить, что на главном окне приложения отображается верное сообщение.







Для проверки того, что нельзя дешифровать сообщение другим ключом, добавим в приложении ещё одного собеседника:

1) Нажать Alt+N для вызова формы добавления нового собеседника.

2) Заполнить поля имени и ключа собеседника.

3) Нажать OK.

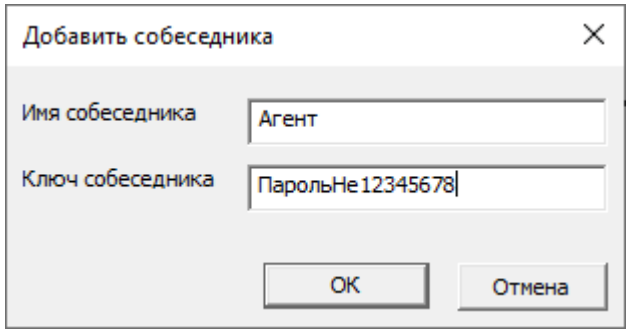
4) Проверить, что на главном окне приложения указан новый собеседник.

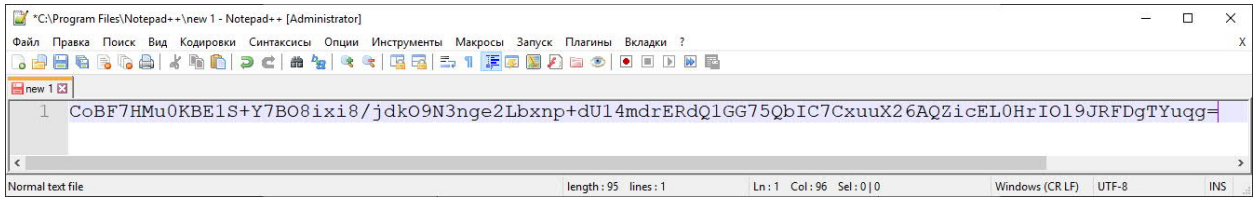
Дешифруем полученное ранее сообщение с использованием ключа нового собеседника.

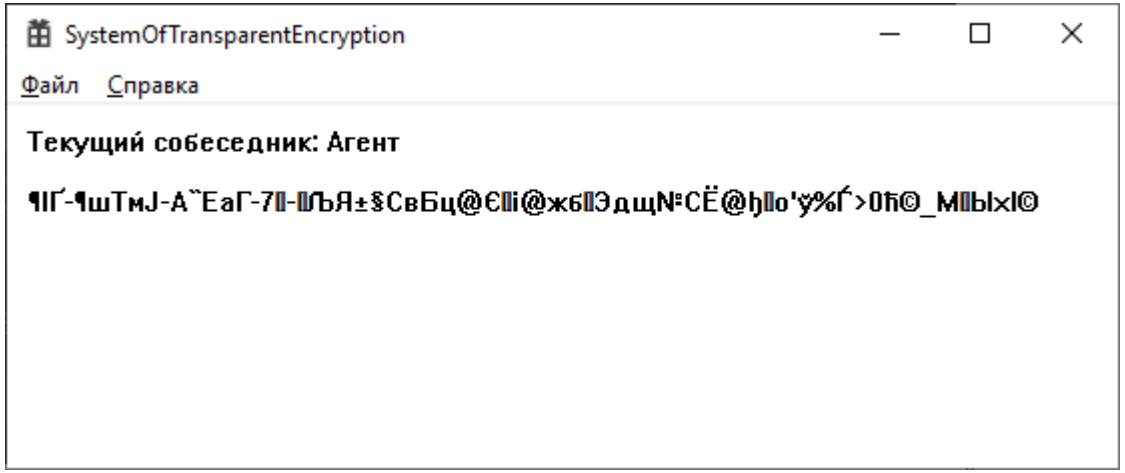
5) Выделить шифротекст.

6) Нажать Alt+D для дешифрования.

7) Проверить, что на главном окне приложения отображается несвязный набор символов.







Наш собеседник отправляет нам фразу: «Отправленное через Telegram сообщение корректно дешифровано», предварительно зашифровав его с помощью разработанного приложения.

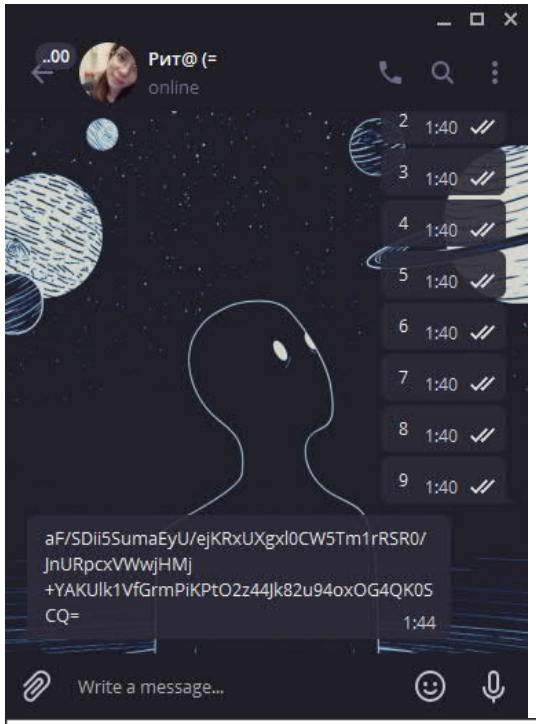
Дешифруем полученное сообщение с помощью следующих действий:

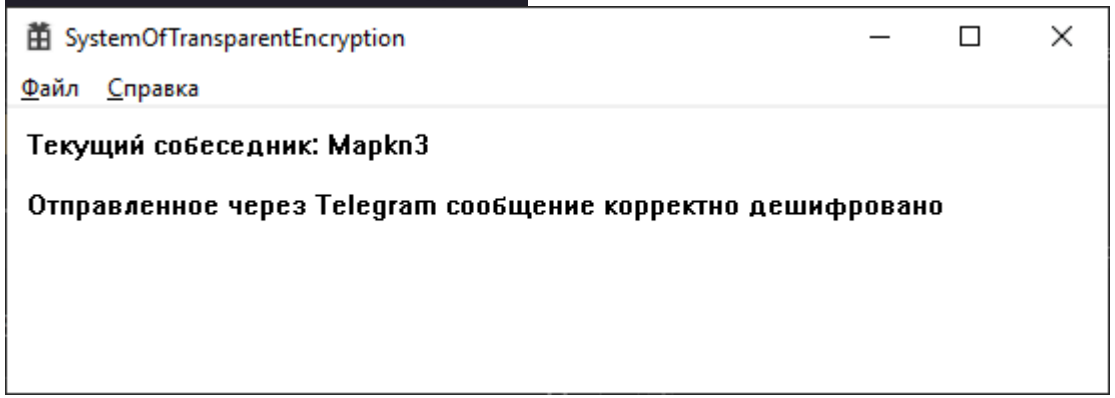
1) Выделить полученное сообщение.

2) Скопировать выделенный текст, нажав Ctrl+C.

3) Нажать Alt+Shift+D для дешифрования теста, находящегося в буфере обмена.

4) Прочитать полученное сообщение в главном окне приложения.

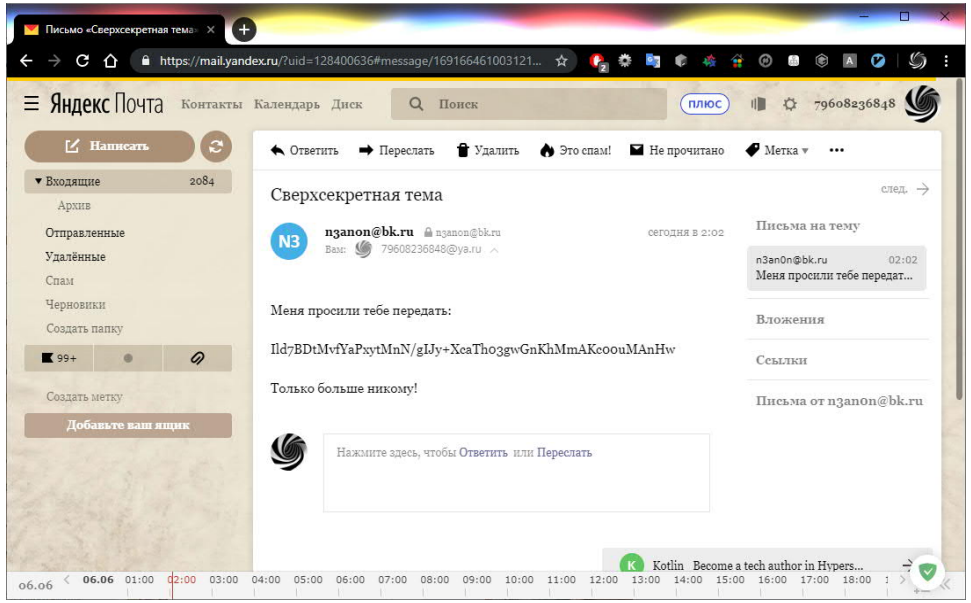




Попросим собеседника отправить на личный электронный адрес фразу: «Передаю привет по E-Mail», предварительно её зашифровав.

Дешифруем аналогичным образом, как описано на предыдущем слайде.

В примере письма показано, что необязательно всё сообщение должно быть зашифровано. Можно посылать смешанные сообщения, состоящие как из зашифрованного, так и из обычного текста. Также, мы можем посылать в одном сообщении несколько отдельных шифрованных фраз. Главное оповестить собеседника о том, как именно отличать разные сообщения друг от друга, например каждую шифрованную фразу начинать с новой строки.



Контрольной фразой будет: «Представляешь, это работает даже в социальных сетях!».

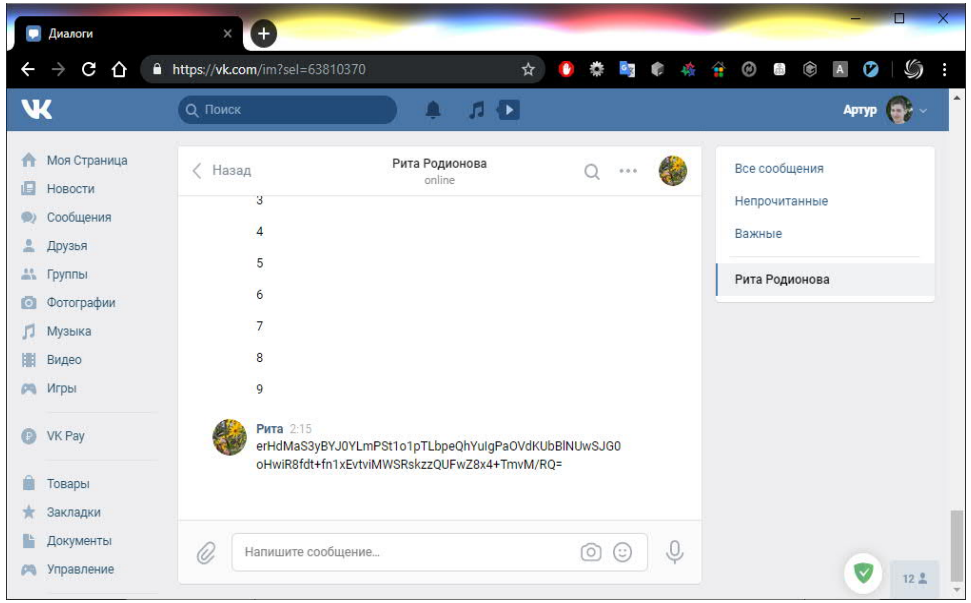
Дешифруем сообщение по следующему алгоритму:

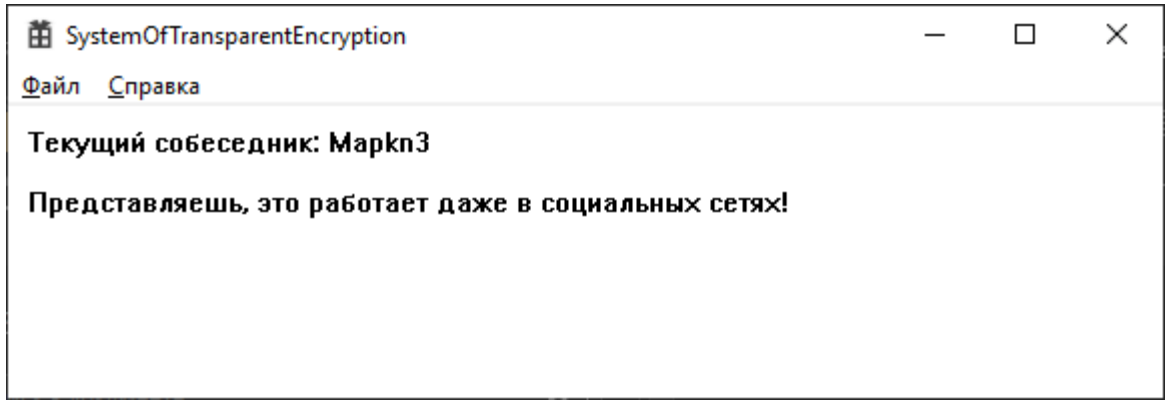
1) Выделить полученное сообщение.

2) Скопировать выделенный текст, нажав Ctrl+C.

3) Нажать Alt+Shift+D для дешифрования теста, находящегося в буфере обмена.

4) Прочитать полученное сообщение в главном окне приложения.





ЗАКЛЮЧЕНИЕ

Проведённый обзор существующих решений в области шифрования передаваемых данных показал отсутствие универсальности и трудность в настройке и эксплуатации. В связи с этим был разработан собственный инструмент шифрования выборочных данных, способный работать с разными приложениями.

Цель работы заключалась в разработке простого в эксплуатации, настраиваемого и универсального приложения, обеспечивающего безопасность выбираемых пользователем данных, передаваемых по разным каналам связи. Универсальность разработанного приложения заключается в возможности защищать данные при передаче с помощью любого средства связи, такого как электронная почта или мессенджер, вне зависимости от наличия в них встроенных средств защиты.

Модульная архитектура упрощает разработку и, в случае необходимости, позволяет переносить приложение на другие платформы. При реализации использовались современные алгоритмы шифрования и генерации ключей, такие как AES и Argon2. Была учтена особенность бинарных данных быть воспринятыми как специальные символы и реализовано дополнительное кодирование Base64. Данная процедура является одним из признаков универсальности, так как позволяет не зависеть от формы представления данных, а работать только с широко распространенным текстовым видом представления и передачи данных.

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

БЛАГОДАРЮ
ЗА ВНИМАНИЕ