1. В современном мире основная часть информации передаётся по сети интернет. В связи с этим многие современные программы поддерживают шифрование «из коробки», но как быть с приложениями, разработчики которых не хотят реализовывать шифрование? В связи с этим была поставлена цель: разработать инструмент шифрования фрагмента текста, выделенного пользователем в стороннем приложении. Для достижения цели были поставлены следующие задачи:

-Проанализировать существующие решения в области шифрования передаваемых данных

-Разработать архитектуру и пользовательский интерфейс приложения

-Реализовать приложение выборочного шифрования

-Проверить функциональные возможности и корректность работы приложения.

1. Для передачи данных возможно использовать различные сервисы, такие как E-Mail, соц. сети, мессенджеры в форме клиентов или веб-версий. {Перечислить со слайда}
2. {Описать со слайда существующие решения}
3. Рассмотрев существующие решения, было решено создать приложение, которое должно уметь шифровать/дешифровать данные, для простоты управления должны использоваться горячие клавиши, должна быть возможность общаться одновременно с несколькими пользователями. Общий принцип работы такой, что программа обрабатывает нажатую горячую клавишу, подготавливает данные для шифрования/дешифрования, шифрует/дешифрует и выполняет финальную обработку данных. {Описать по слайду взаимодействие модулей}
4. Для шифрования данных используем алгоритм AES, так как он является стандартом шифрования, изучен экспертами и везде используется. В качестве ключей шифрования будет использовать сеансовые ключи, потому что при шифровании большого объёма данных одним и тем же ключом, злоумышленнику будет легче вычислить этот ключ, используя тот же статистический метод. Для генерации сеансовых ключей будем использовать функцию Argon2. Это функция с открытым исходным кодом, которая в данный момент широко распространена и является рекомендованной к использованию при хешировании паролей. Особенностью Argon2 является его свойство затрачивать не менее определенного значения времени и памяти во время хеширования, что невозможно ускорить и делает атаки полного перебора бессмысленными. Для генерации сеансового ключа будет использовать долговременный ключ, называемый «общим секретом» собеседников, и одноразовое случайное значение Nonce. Чтобы не вводить при каждом запуске программы общие секреты каждого из собеседников, будем хранить их в зашифрованном файле на компьютере пользователя. Ключом для файла будет являться пароль пользователя, который он вводит при запуске приложения. Также, полученные бинарные данные мы кодируем с помощью Base64, чтобы сторонние программы не восприняли их как спец. символы.
5. Для получения и отправки данных пользователя в программе используется буфер обмена. Для того, чтобы не нарушить работу других приложений, перед получением данных мы сохраняем текущее состояние буфера обмена и после отправки восстанавливаем, что было сохранено. Таким образом работу алгоритма шифрования можно описать следующим образом: {Описать блок-схемы на слайде}. Шифрование можно описать следующими формулами, где {Описать формулы}. И формат сообщения, сгенерированного программой, будет следующим: {Описать формат сообщения}. Алгоритм дешифрования проводится аналогичным образом, но дешифрованный текст не вставляется на место выделенного, а появляется на главном окне приложения.
6. {Следующие слайды описывают проверку работоспособности приложения}
7. 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. 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. 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.