Версия 1.0 от 2017-07-27

Приложение «SLS PGP»

Руководство пользователя

Листов 9

Москва, 2017

ОГЛАВЛЕНИЕ

1. АННОТАЦИЯ

Данный документ содержит описание приложения для шифрования файлов "SLS PGP".

Приложение "SLS PGP" представляет собой JAR-файл, предназначенный для выполнения в JRE. Для запуска приложения требуется предустановленный JRE версии не менее 1.8.

Приложение "SLS PGP" предназначено для шифрования и расшифровывания файлов. С его помощью можно обеспечить безопасный обмен данными по незащищённым каналам связи.

Приложение генерирует ключевые пары RSA, с помощью которых производится шифрование и расшифровывание файлов. Файл-контейнер с секретной частью ключевой пары защищается паролем.

В процессе шифрования используются криптографические алгоритмы RSA, SHA-256, AES.

В документе приводится необходимая справочная информация для работы с приложением.

2. ТЕРМИНОЛОГИЯ И УСЛОВНЫЕ ОБОЗНАЧЕНИЯ (ГЛОССАРИЙ)

JRE – Java Runtime Environment, среда выполнения для Java. Требуется для запуска Java-приложений.

JAR – Java Archive, ZIP-архив, в котором содержится программа (или её часть) на языке Java.

RSA – криптографический асимметричный алгоритм с открытым ключом.

AES – криптографический симметричный алгоритм блочного шифрования.

SHA-256 – криптографический алгоритм хеширования.

3. АЛГОРИТМ РАБОТЫ С ПРИЛОЖЕНИЕМ

3.1 Генерация ключевой пары

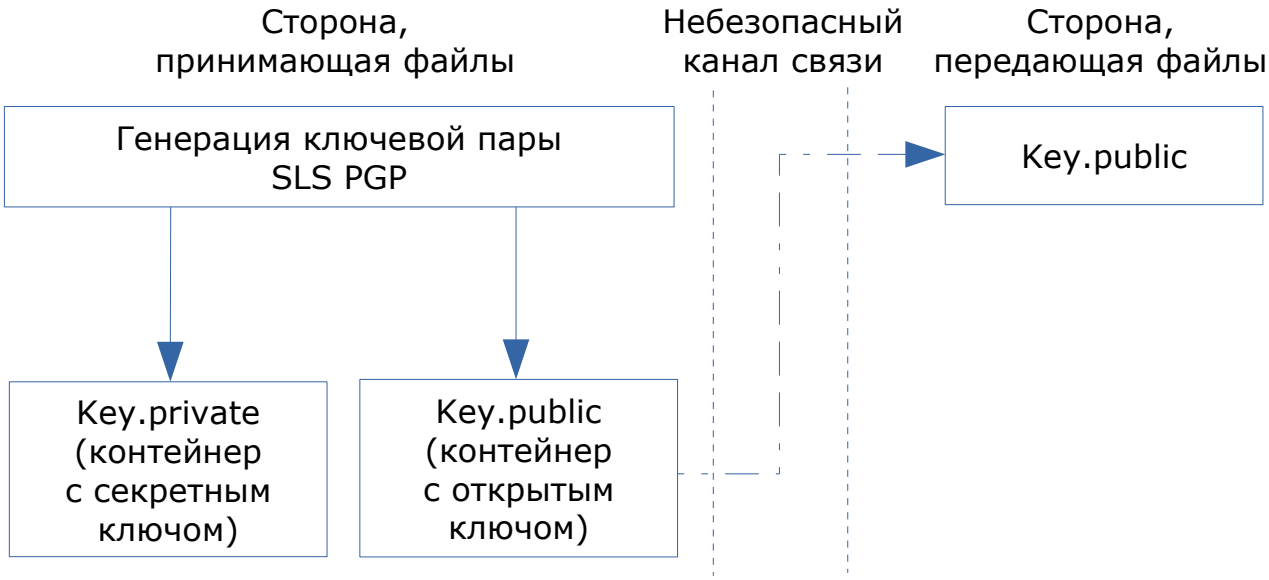
Первым этапом работы с приложением "SLS PGP" является генерация ключевой пары (см. раздел 4.1). Сгенерированная ключевая пара существует в виде двух контейнеров, содержащих соответственно открытый и секретный ключи.

Шифрование файлов производится с использованием открытого ключа. Расшифровка файлов производится с использованием секретного ключа. Только владелец секретного ключа (соответствующего использованному открытому ключу) может расшифровать файлы, зашифрованные приложением.

Важно! Файл-контейнер с секретным ключом должен храниться в защищенном месте, доступном только владельцу ключа.

3.2 Передача контейнера с открытым ключом

Файл-контейнер с открытым ключом передается всем, кто хочет безопасно передавать владельцу секретного ключа какие-либо файлы по небезопасным каналам связи.



Для предотвращения атак типа "человек посередине" рекомендуется дополнительно удостовериться, что контейнер с открытым ключом был действительно передан тем человеком, которому будут предназначаться впоследствии зашифрованные файлы. Для этого можно, например, связаться по телефону и сверить часть данных переданного контейнера.

3.3 Передача файла в зашифрованном виде

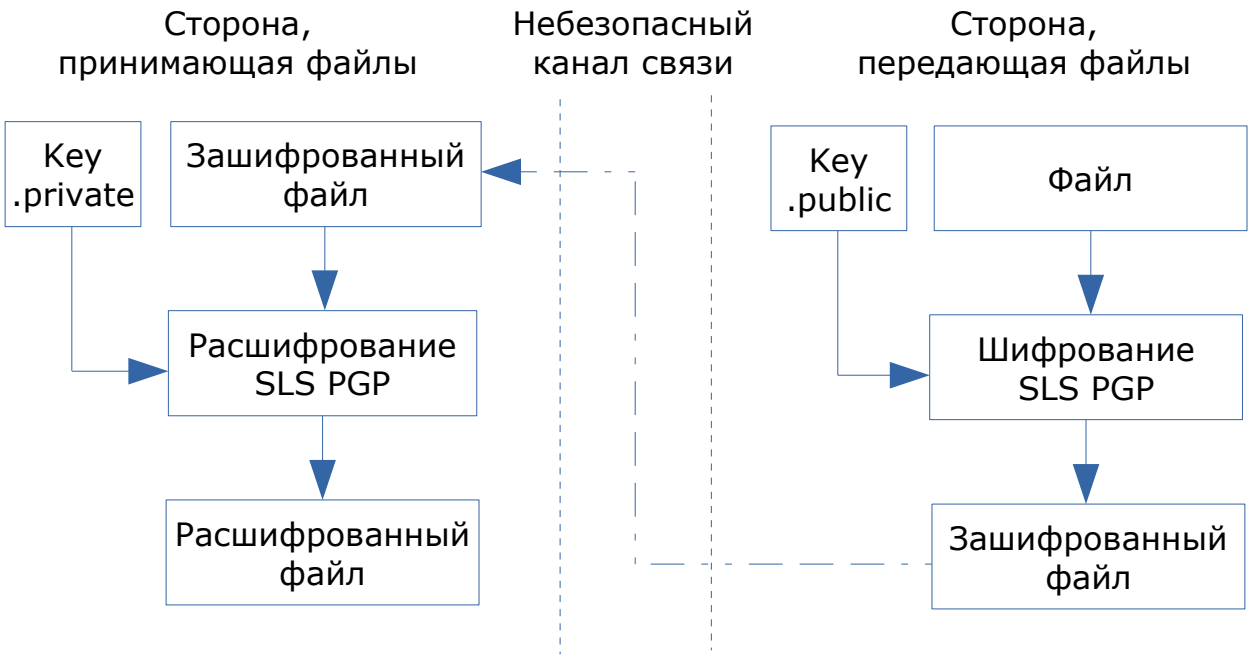
После успешной передачи другой стороне контейнера с открытым ключом, можно осуществлять передачу файлов по небезопасным каналам связи.

Для этого передающая сторона:

- Зашифровывает файл, используя контейнер с открытым ключом (см. раздел 4.2)
- Передает зашифрованный файл по небезопасному каналу связи

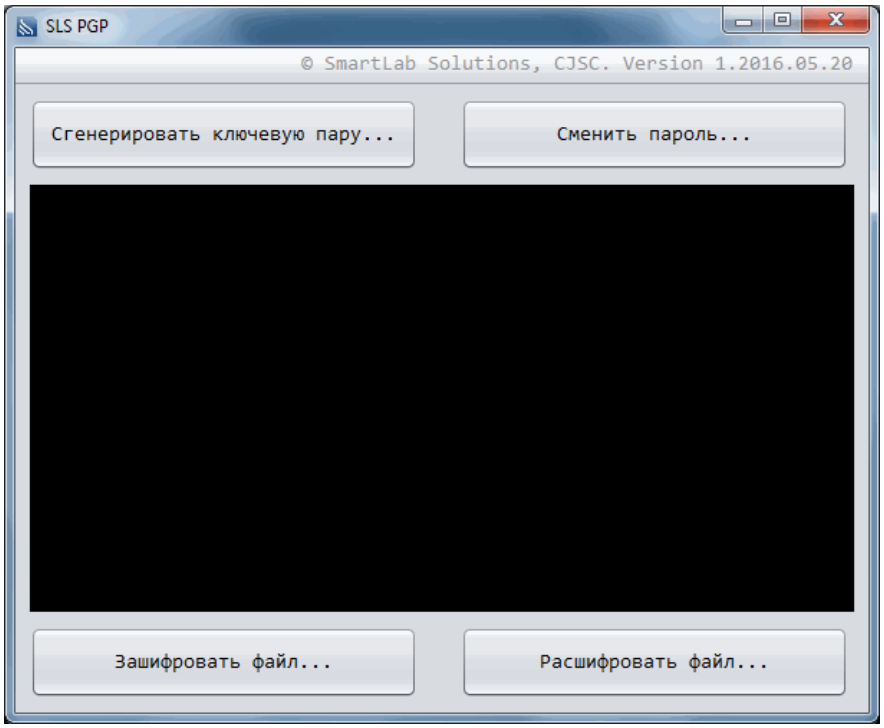
Принимающая сторона:

- Получает зашифрованный файл
- Расшифровывает полученный файл, используя контейнер с секретным ключом (см. раздел 4.3)



4. РАБОТА С ПРИЛОЖЕНИЕМ

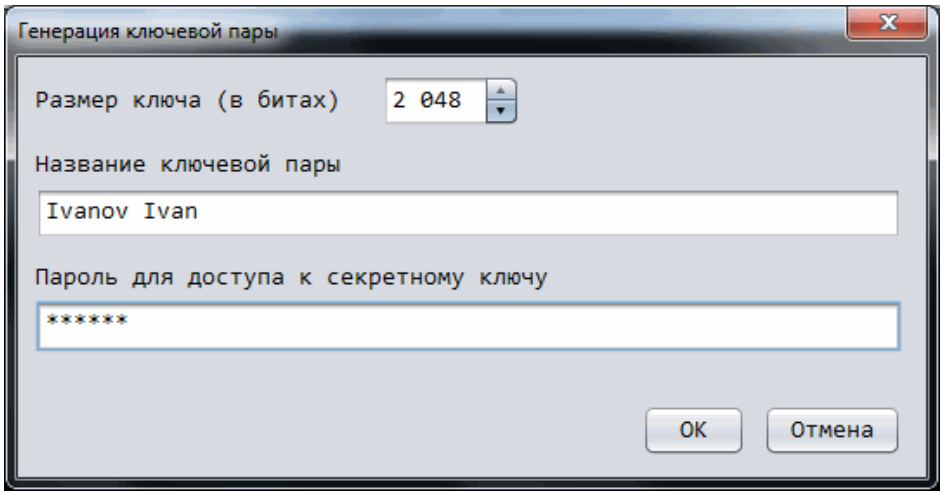
Окно приложения после запуска выглядит следующим образом:



Приложение содержит 4 кнопки и лог вывода сообщений о результатах выполнения выбранных операций. Описание функционала приложения даётся ниже.

4.1 Генерация ключевой пары

При нажатии кнопки "Сгенерировать ключевую пару...", выводится диалог задания параметров ключевой пары. Окно диалога выглядит следующим образом:



Название ключевой пары должно состоять только из символов, допустимых в именах файлов в операционной системе пользователя.

Пароль доступа к секретному ключу не должен быть пустым и может содержать любые допустимые в операционной системе пользователя символы.

Размер ключа влияет на надёжность шифрования файлов с использованием данной ключевой пары.

После нажатия кнопки "ОК" (если все параметры заданы корректно) будет выведен диалог выбора папки для сохранения ключевой пары.

Далее начнётся генерация ключевой пары. Время генерации ключевой пары зависит от размера ключа. Например, генерация ключевой пары с размером ключа 2048 бит занимает несколько секунд, с размером ключа 8192 бит – несколько часов.

Если генерация ключевой пары с именем [*имя_ключевой_пары*] прошла успешно, в лог приложения выводится соответствующее сообщение. В выбранной папке будут созданы два файла:

- [*имя_ключевой_пары*].private — контейнер с секретным ключом (секретный ключ хранится в зашифрованном виде, защищённый паролем)
- [*имя_ключевой_пары*].public — контейнер с открытым ключом (открытый ключ хранится в виде текста)

4.2 Шифрование файла

При нажатии кнопки"Зашифровать файл...", последовательно выводятся диалоги:

- Выбор файла для шифрования
- Выбор контейнера с открытым ключом

При успешном выборе начинается шифрование. Время шифрования пропорционально размеру файла.

Если шифрование файла прошло успешно, в лог приложения выводится соответствующее сообщение. В папке с исходным файлом [*имя_файла*] создаётся зашифрованный файл [*имя_файла*].enc.

4.3 Расшифрование файла

При нажатии кнопки "Расшифровать файл...", последовательно выводятся диалоги:

- Выбор файла для расшифровывания
- Выбор контейнера с секретным ключом
- Ввода пароля для контейнера с секретным ключом

При успешном выборе и корректном пароле начнется расшифровывание файла. Время расшифровывания пропорционально размеру файла.

Если расшифровывание файла прошло успешно, в лог приложения выводится соответствующее сообщение. В папке с файлом [*имя_файла*].enc создаётся расшифрованный файл [*имя_файла*].

4.4 Смена пароля контейнера с секретным ключом

Чтобы сменить пароль для доступа к контейнеру с секретным ключом (например, если пароль был скомпрометирован), необходимо нажать на кнопку "Сменить пароль…". Последовательно выводятся диалоги:

- Выбор контейнера с секретным ключом
- Ввод старого пароля для доступа к контейнеру
- Ввод нового пароля

Пароль доступа к секретному ключу не должен быть пустым и может содержать любые допустимые в операционной системе пользователя символы.

Если смена пароля прошла успешно, в лог приложения выводится соответствующее сообщение.