Лабораторная работа по криптографии №3

Гибридная криптосистема, реализованная с использованием библиотек .NET

Автор: Гашков С.П., гр.22305

23.02.2024

1. Формулировка задания

При помощи функций криптографической библиотеки .NET реализуйте гибридную

криптосистему, включающую:

1) генерацию ключевой пары RSA;

2) шифрование и расшифрование документа симметричным криптоалгоритмом;

3) шифрование и расшифрование сеансового ключа симметричного алгоритма при

помощи ключей RSA;

4) формирование и проверку цифровой подписи документа.

Полученный шифротекст, открытые ключи должны сохраняться и передаваться

через файлы.

1. Описание метода решения

Система поддерживает 2 режима работы:

***Шифрование:***  
1. Пользователь вводит имя и адрес документа, который необходимо зашифровать.

2. Генерируется симметричный ключ для алгоритма AES (состоит из двух компонент: Key и IV)[[1]](#footnote-1)

3. Документ шифруется симметричным ключом AES. Зашифрованный документ сохраняется в виде файла encryptedDocument.txt

4. Генерируется сессионные открытый и закрытый ключи для алгоритма RSA. Записываются, соответственно, в publicKeyRSA.xml и privateKeyRSA.xml

5. Симметричный ключ AES шифруется алгоритмом RSA и записывается в файл encryptedAESKey.xml.

6. На основе открытого ключа RSA генерируется цифровая подпись для зашифрованного документа и сохраняется в digitalSignatureFromEncryptedDocument.txt.

***Дешифрование:***

1. Пользователь вводит имя и адрес цифровой подписи зашифрованного документа. Если подписи не существует, система требует повторный ввод имени и адреса цифровой подписи.[[2]](#footnote-2)

2. Пользователь вводит имя и адрес зашифрованного документа. Если документа не существует, система требует повторный ввод имени и адреса зашифрованного документа.

3. Пользователь вводит имя и адрес сессионного открытого ключа RSA. Если ключа не существует, система требует повторный ввод имени и адреса зашифрованного ключа.

4. Проверка правильности цифровой подписи. О результатах проверки сообщается пользователю.

5. Пользователь вводит имя и адрес зашифрованного симметричного ключа AES. Если ключа не существует, система требует повторный ввод имени и адреса ключа.

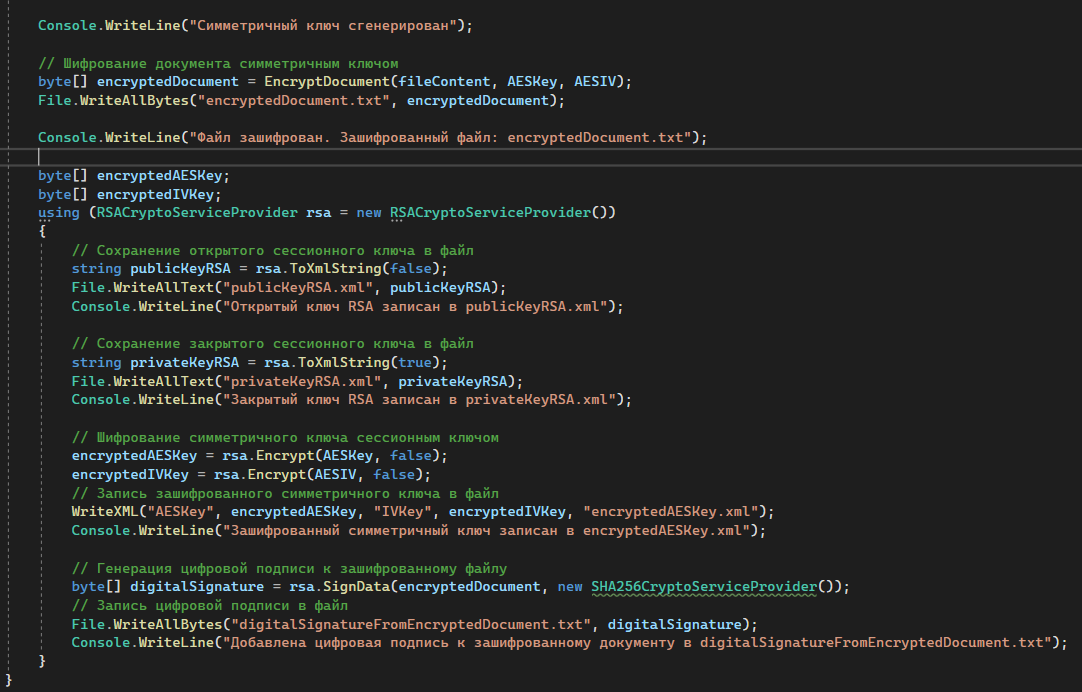
6. Пользователь вводит имя и адрес сессионного закрытого ключа RSA. Если ключа не существует, система требует повторный ввод имени и адреса ключа.

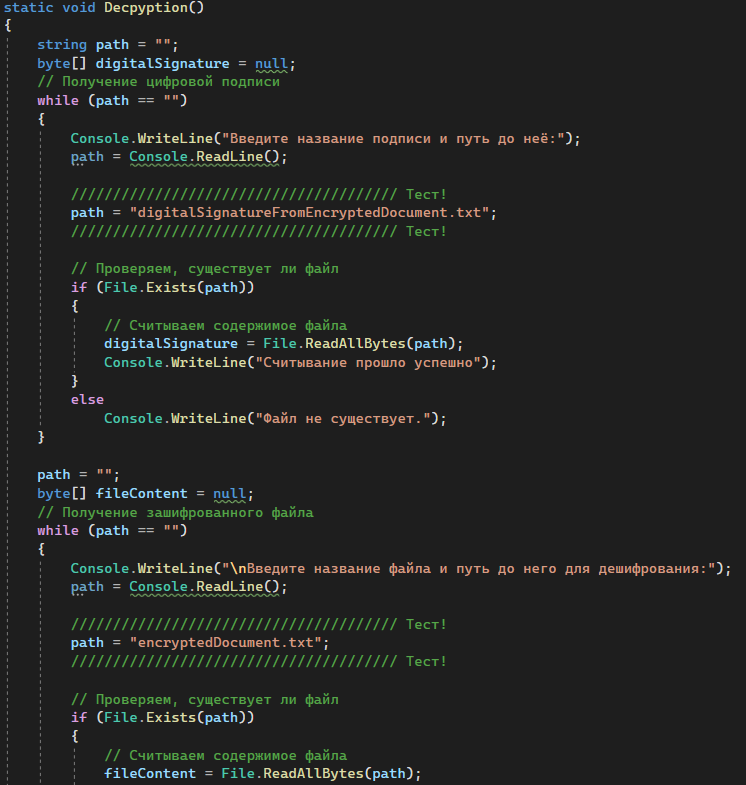
7. Дешифровка симметричного ключа AES закрытым ключом RSA.

8. Дешифровка текста симметричным ключом AES и запись дешифрованного текста в DecryptedDocument.txt.

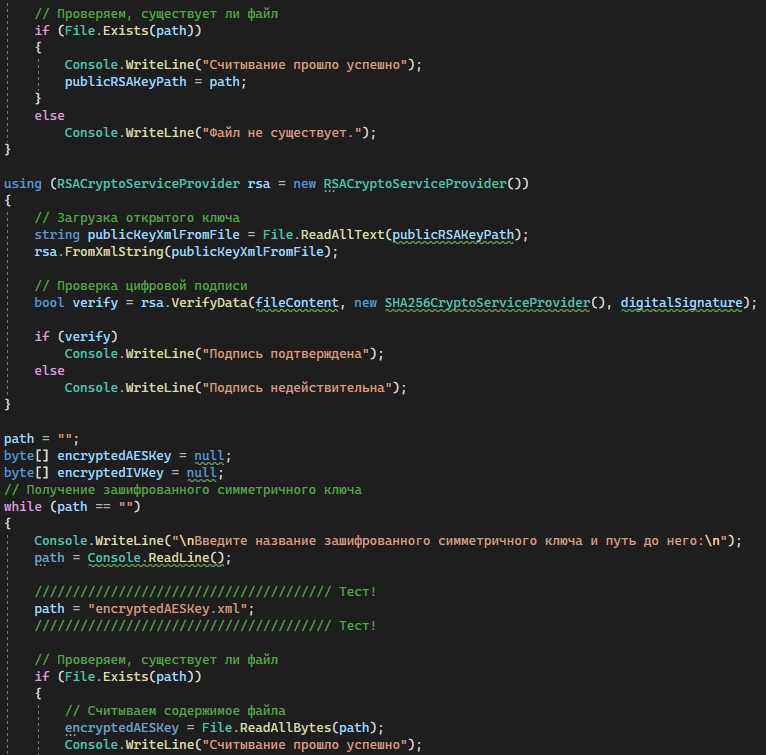
1. Примеры кода

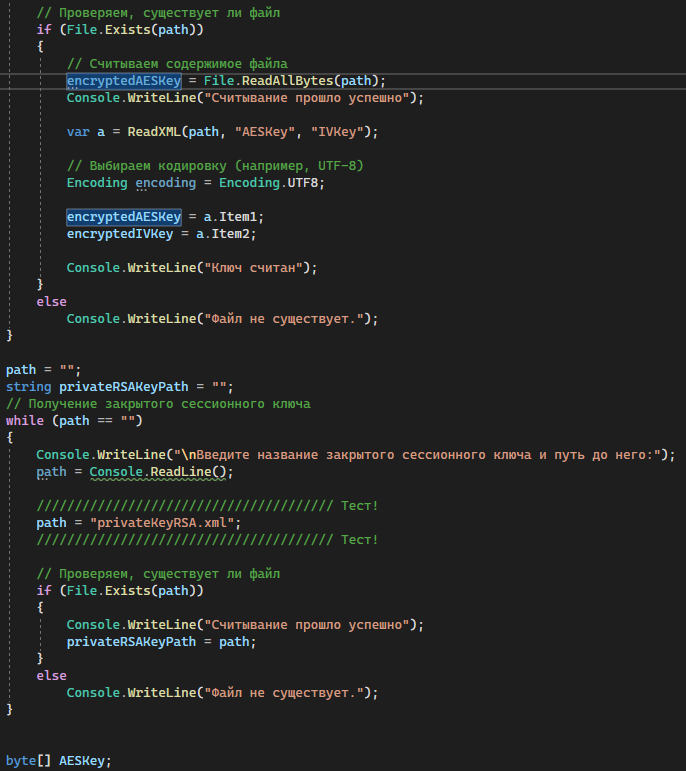











1. Тестовые данные
   1. Сравнение текстов в example.txt и DecryptedDocument.txt – позитивный тест
   2. Сравнение текстов в example.txt и encryptedDocument.txt – негативный тест
2. Ссылка на репозиторий

<https://github.com/Nemo121007/cryptography.git>

1. Здесь и далее для генерации ключей, шифрования, дешифрования и т.д. используются готовые функции платформы .NET (главным образом библиотека System.Security.Cryptography) [↑](#footnote-ref-1)
2. Здесь и далее при разработке по-умолчанию обязательно устанавливаются имена файлов, совпадающие с файлами, записанными в ходе шифрования. Для изменения обязательной установки предустановленных имён удалить присваивание переменной path фиксированного значения (также данное присваивание выделено метками «Тест») [↑](#footnote-ref-2)