Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Практическое задание №8**

**Тема «Изучение стандартных средств для реализации приложений, использующих симметричное и ассиметричное шифрование с использованием библиотеки System.Security.Cryptography»**

Руководитель: Ржеутская Н. В.

Выполнил:

Студент 2 курса 1 группы ФИТ

Немкович Анастасия Вадимовна

Минск 2023

**Цель работы**

Изучить модель криптографии .NET Framework, Основные классы и структуры данных, разработать приложение для шифрования файлов использующих симметричные и ассиметричные алгоритмы шифрования.

**Теоретические сведения**

В .Net Framework присутствует пространство имён для выполнения криптографических операций под названием System.Security.Cryptography. Данное пространство имён предоставляет криптографические службы, включающие безопасное кодирование и декодирование данных, а также другие операции, такие как хеширование сообщений, генерация случайных чисел и проверка подлинности сообщений. Данная библиотека предоставляет доступ для использования различных реализаций алгоритмов в основном это программные интерфейсы CryptoApi (CAPI) и Cryptography Next Generation API (CNG API) помимо этого для некоторых алгоритмов возможно использование реализаций на основе OpenSsl.

CryptoAPI — интерфейс программирования приложений, который обеспечивает разработчиков Windows-приложений стандартным набором функций для работы с криптопровайдером. Входит в состав операционных систем Microsoft. Большинство функций CryptoAPI поддерживается, начиная с Windows 2000.

Cryptography Next Generation стала долгосрочной заменой CAPI. Данный набор интерфейсов поддерживает все алгоритмы предлагаемые CAPI а также другие алгоритмы перечисленные в своде правил Suite B Агентства национальной безопасности США [1]. Данный интерфейс поддерживает следующие длины ключей или размерность хеша.

• RSA 512 бит to 16384 бит, с шагом 64 бит

• DH 512 бит to 16384 бит, с шагом 64 бит

• DSA 512 бит to 1024 бит, с шагом 64 бит

• ECDSA P-256, P-384, P-521 (NIST Curves)

• ECDH P-256, P-384, P-521 (NIST Curves)

• MD2 128 бит

• MD4 128 бит

• MD5 128 бит

• SHA-1 160 бит

• SHA-256 256 бит

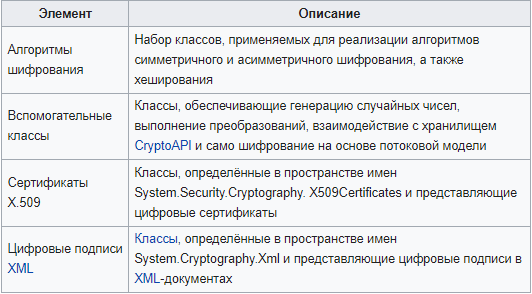
• SHA-384 384 бит

• SHA-512 512 бит

Рассматривая структуру наследования для симметричных алгоритмов в .Net стоит упомянуть что SymmetricAlgorithm является абстрактным классом, от который наследуют абстрактные классы для реализаций каждого из алгоритмов. В свою очередь каждая из реализации алгоритма является производной от абстрактного класса алгоритма.

Основное назначение библиотеки System.Security.Cryptography.

[.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) включает набор криптографических сервисов, расширяющих аналогичные сервисы [Windows](https://ru.wikipedia.org/wiki/Windows) через [CryptoAPI](https://ru.wikipedia.org/wiki/CryptoAPI). [Пространство имен](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%B8%D0%BC%D0%B5%D0%BD" \o "Пространство имен) System.Security.Cryptography открывает программный доступ к самым разнообразным криптографическим сервисам, с помощью которых приложения могут шифровать и дешифровать [данные](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5), обеспечивать их целостность, а также обрабатывать цифровые подписи и сертификаты.



Размер ключа измеряется в битах (двоичных разрядах). Чем он больше, тем, соответственно, больше времени необходимо на перебор возможных значений, но и тем продолжительнее работает алгоритм. Поэтому выбор оптимальной длины ключа – это вопрос баланса.

Существует множество (не менее двух десятков) [алгоритмов](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) симметричных шифров, существенными параметрами которых являются:

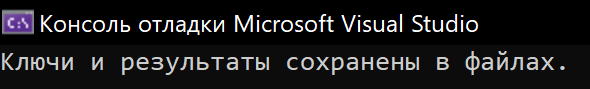
* [стойкость](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D1%82%D0%BE%D0%B9%D0%BA%D0%BE%D1%81%D1%82%D1%8C)
* длина ключа
* число [раундов](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%83%D0%BD%D0%B4_(%D0%B2_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%B8))
* длина обрабатываемого блока
* сложность аппаратной/программной реализации
* сложность преобразования

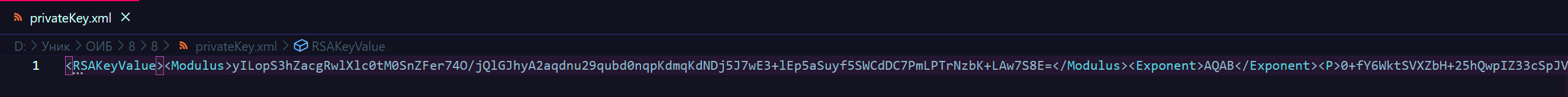
**Ход работы**

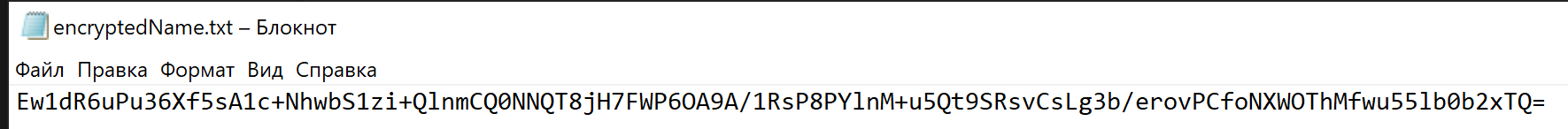
1. Ознакомиться с созданием криптографического приложения;

|  |  |  |
| --- | --- | --- |
| № | Алгоритм шифрования(размер ключа) | Алгоритм хеширования |
| 12 | RSA (640bit) | SHA384 |

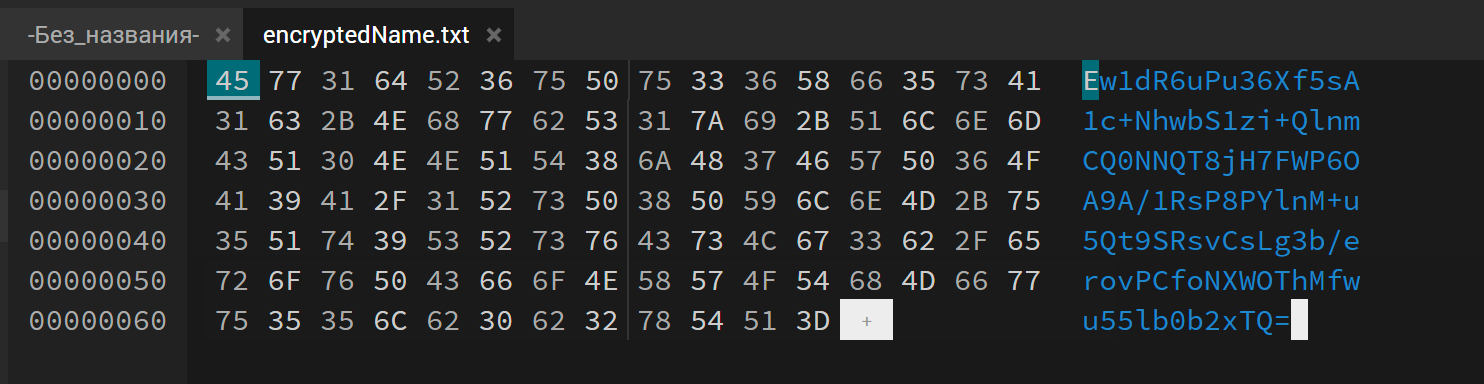
1. Выполнить шифрование, дешифрование и хеширование своей фамилии по указанным алгоритмам. Используя функции работы с файлами сохранить ключи шифрования, результаты шифрования и хеширования.

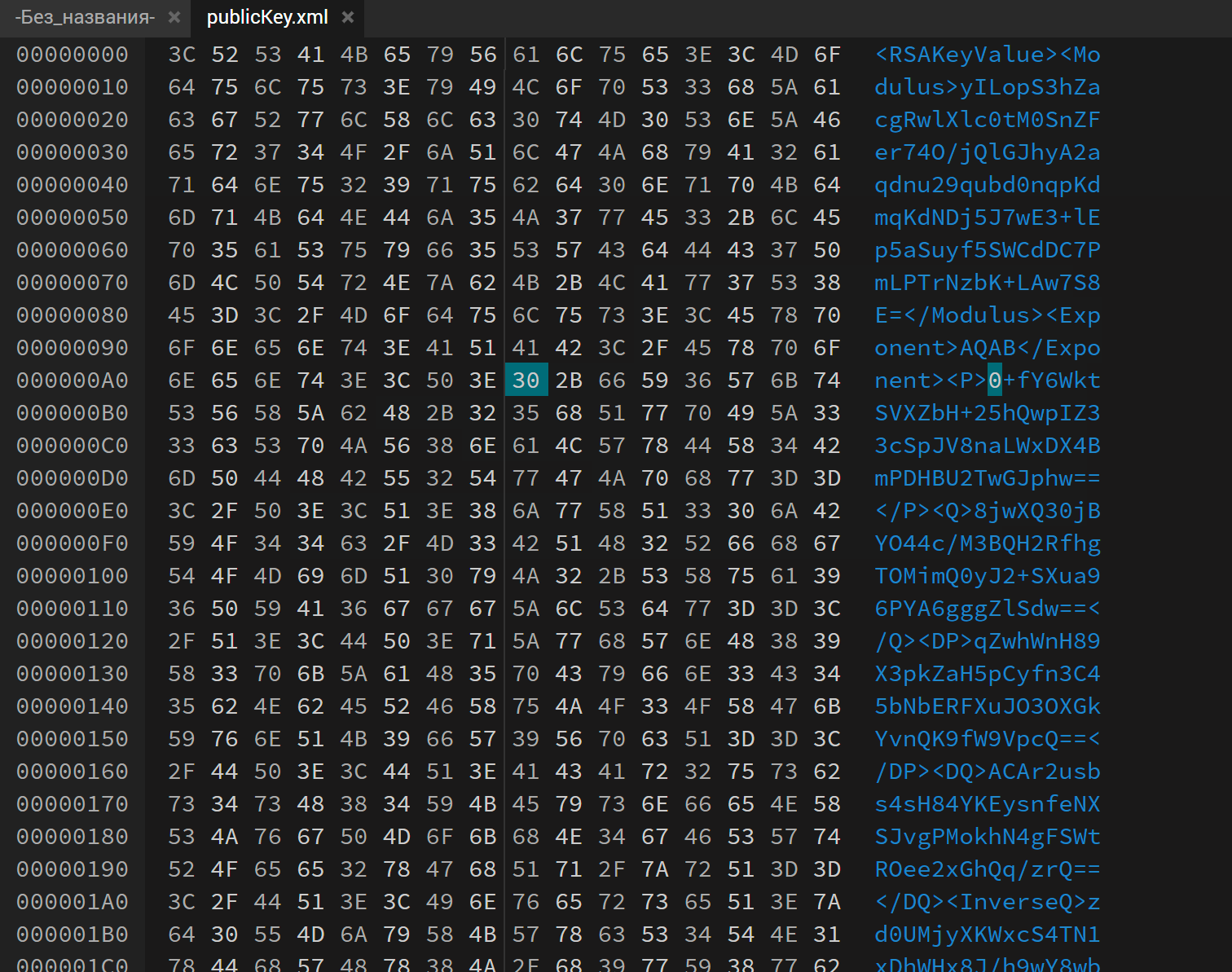


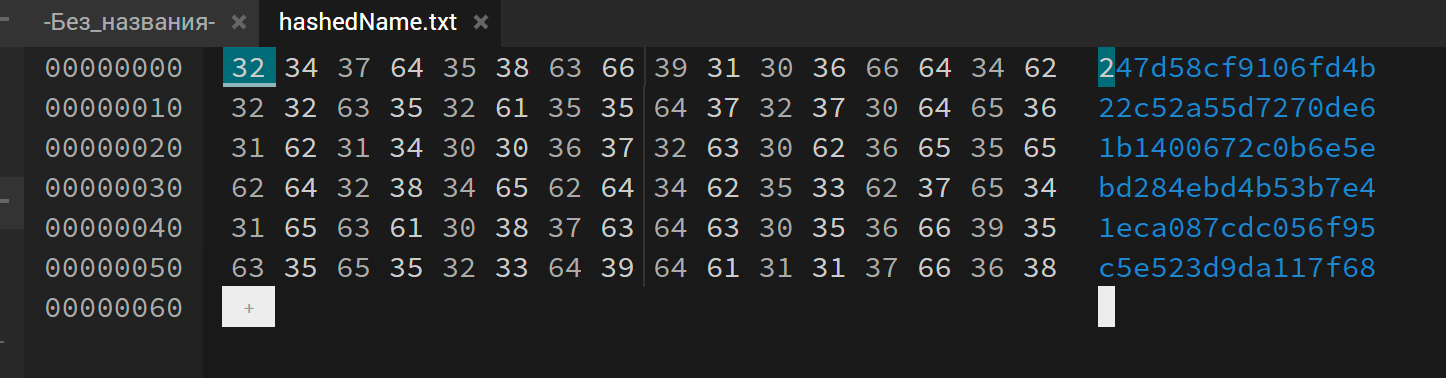


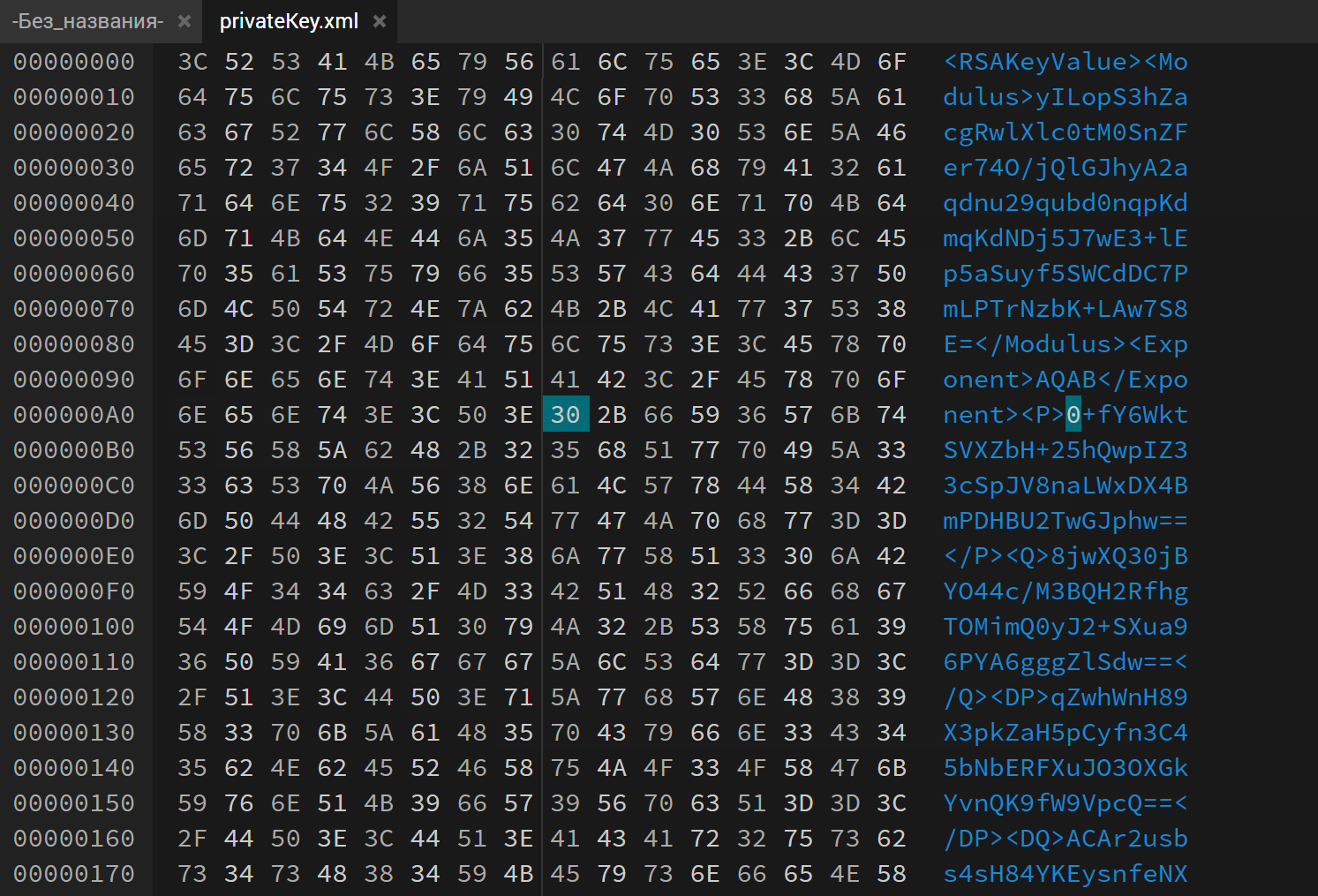


1. Для выше указанных алгоритмов используя Hex-редактор продемонстрировать ключи шифрования, зашифрованные и захешированные данные.

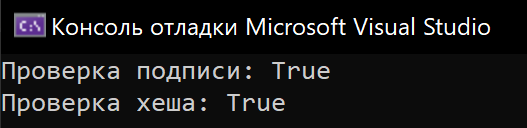


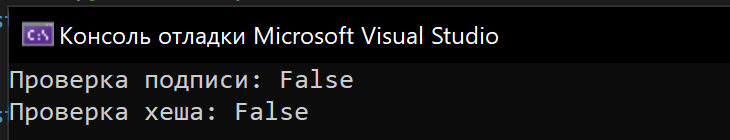






1. \*Реализовать проверку сообщения (фамилии) и хеша по примеру ЭЦП. Также продемонстрировать что будет если будет изменен хеш или сообщение. Пример того что нужно реализовать представлен ниже.





**Вывод**

Под криптографической защитой информации понимается такое преобразование исходной информации, в результате которого она становится недоступной для ознакомления и использования лицами, не имеющими на это полномочий.

[System.Security.Cryptography](https://docs.microsoft.com/ru-ru/dotnet/api/system.security.cryptography) предоставляет криптографические услуги, включая безопасное кодирование и декодирование данных, а также многие другие операции, такие как хеширование, генерация случайных чисел и аутентификация сообщений.