ПРАКТИЧНА РОБОТА №6  
З ОСНОВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ  
СТУДЕНТА КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКО  
ФАКУЛЬТЕТУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРИ МЕРЕЖЕВИХ ТА ІНТЕРНЕТ ТЕХНОЛОГІЙ  
ДРУГОГО КУРСУ, ДРУГОЇ ПІДГРУПИ  
ОЛЕКСАНДРА ОЛЕКСАНДРОВИЧА ФУРКАЛА  
ЗА ТЕМОЮ “ЕЛЕКТРОНИЙ ЦИФРОВИЙ ПІДПИС”  
ЗВІТ  
20.11.2022

Завдання цієї практичної роботи були такі:  
1. Написати програму, яка виконує підписування повідомлення та   
перевірку ЕЦП з використанням алгоритму асиметричного шифрування RSA. Відкритий ключ імпортується із XML-файлу, секретний – береться із контейнера.  
2. Оформити звіт.

Важливою функцією криптографії є ​​забезпечення неспростовності надісланого повідомлення, коли одержувач повідомлення не може заперечити, що повідомлення є автентичним. Цифровий підпис — це техніка, яка використовується для демонстрації автентичності повідомлення. Цифровий підпис дає одержувачу підпису підстави вважати, що ідентифікований відправник створив вихідне повідомлення.  
Електронний цифровий підпис – це засіб, що дозволяє на основі криптографічних методів надійно встановити авторство і справжність електронного документа. Електронний цифровий підпис дозволяє замінити при безпаперовому документообігу традиційні печатку та підпис. Проставляння підпису під документом не змінює самого документа, а тільки дає можливість перевірити достовірність та авторство отриманої інформації. Таким чином, досягається автентичність (справжність) та цілісність повідомлення. З юридичної точки зору це означає, що автор повідомлення не зможе від нього відмовитися.

Вимоги до ЕЦП (тобто Електронного цифрового підпису):  
- Має доводити, що саме законний автор, і ніхто інший, свідомо підписав документ;  
- Має представляти собою невід’ємну частину документа. Має бути неможливим відокремлення підпису від документа і використання його для підписання інших документів;  
- Повинен забезпечувати неможливість зміни підписаного документа (в тому числі і для самого автора);  
- Факт підписання документа повинен бути юридично-доказовим (забезпечення автентичності)

Цифрові підписи засновані на асиметричній криптографії (хоча спочатку були спроби створення цього механізму на основі алгоритмів симетричного шифрування). Для одержувача повідомлення цифровий підпис дозволяє одержувачу вірити, що повідомлення надіслав правильний відправник, це можна вважати цифровим еквівалентом підпису на листі, але цифровий підпис набагато важче підробити.   
Цифрові підписи забезпечують як автентифікацію, так і неспростовність. Автентифікація, оскільки підписи мають бути створені користувачем із дійсним приватним ключем, і неспростовність, оскільки одержувач може довіряти, що відомий відправник підписав повідомлення, оскільки лише він знає закритий ключ. Лише збереження у таємниці секретного ключа гарантує неможливість (точніше – велику обчислювальну складність) підробки зловмисником документа і цифрового підпису від імені автора.   
Відкритий ключ підпису формується як значення деякої функції від секретного ключа, але знання відкритого ключа не дозволяє визначити секретний ключ. Відкритий ключ може бути опублікований і використовується для перевірки достовірності підписаного документа, а також для попередження шахрайства з боку автора у вигляді відмови його від підписаного ним документа.

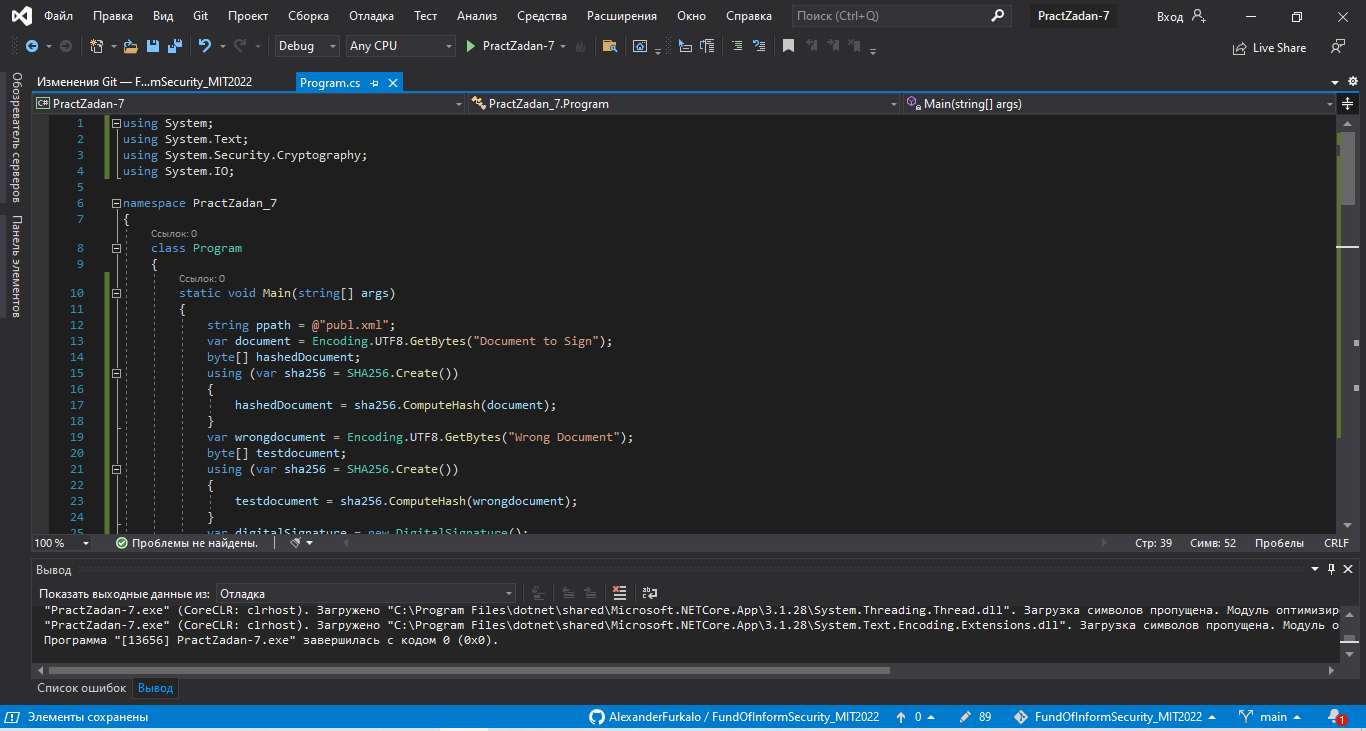
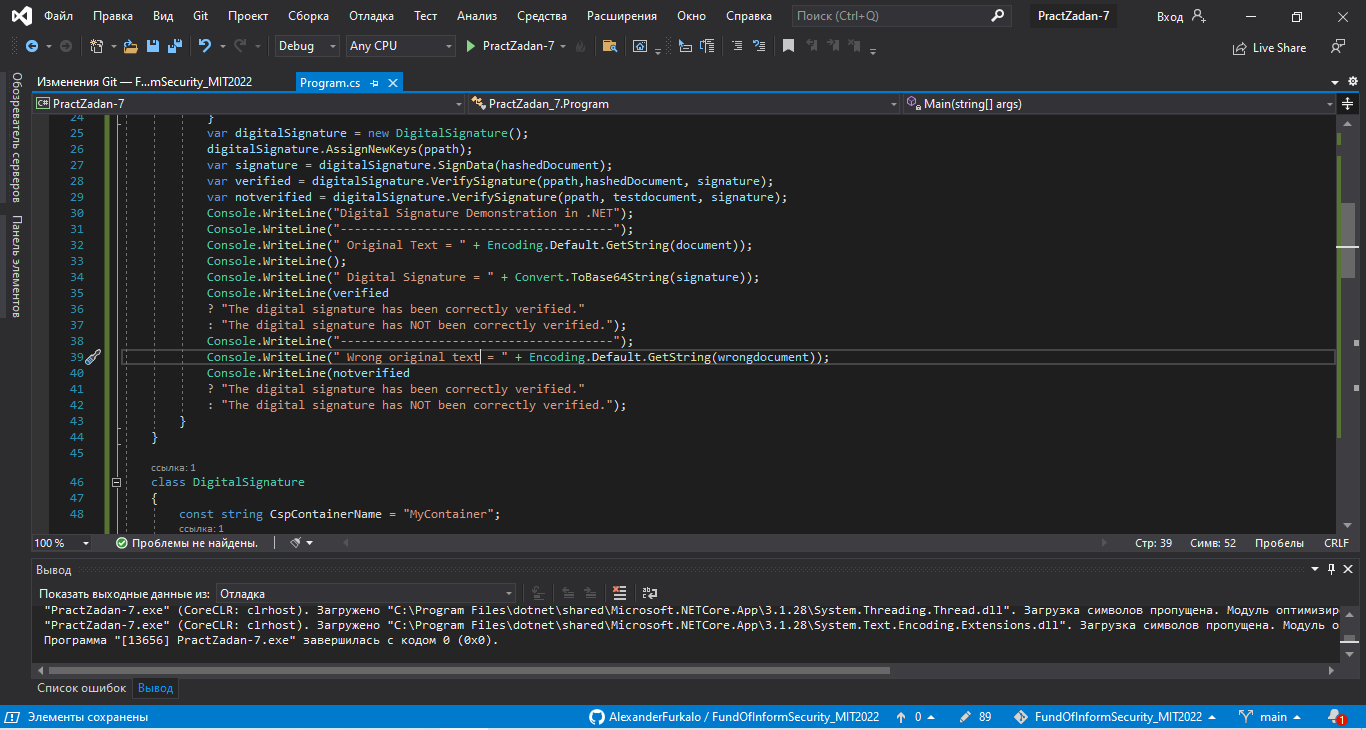
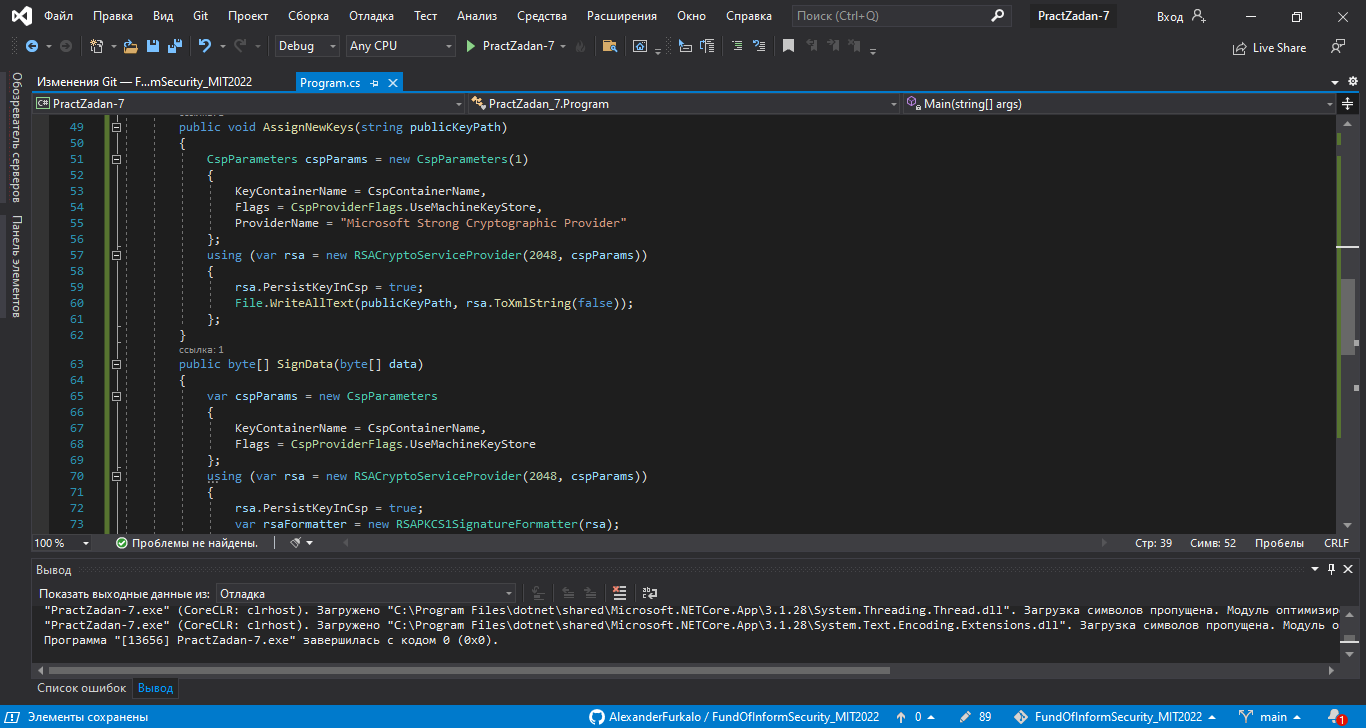
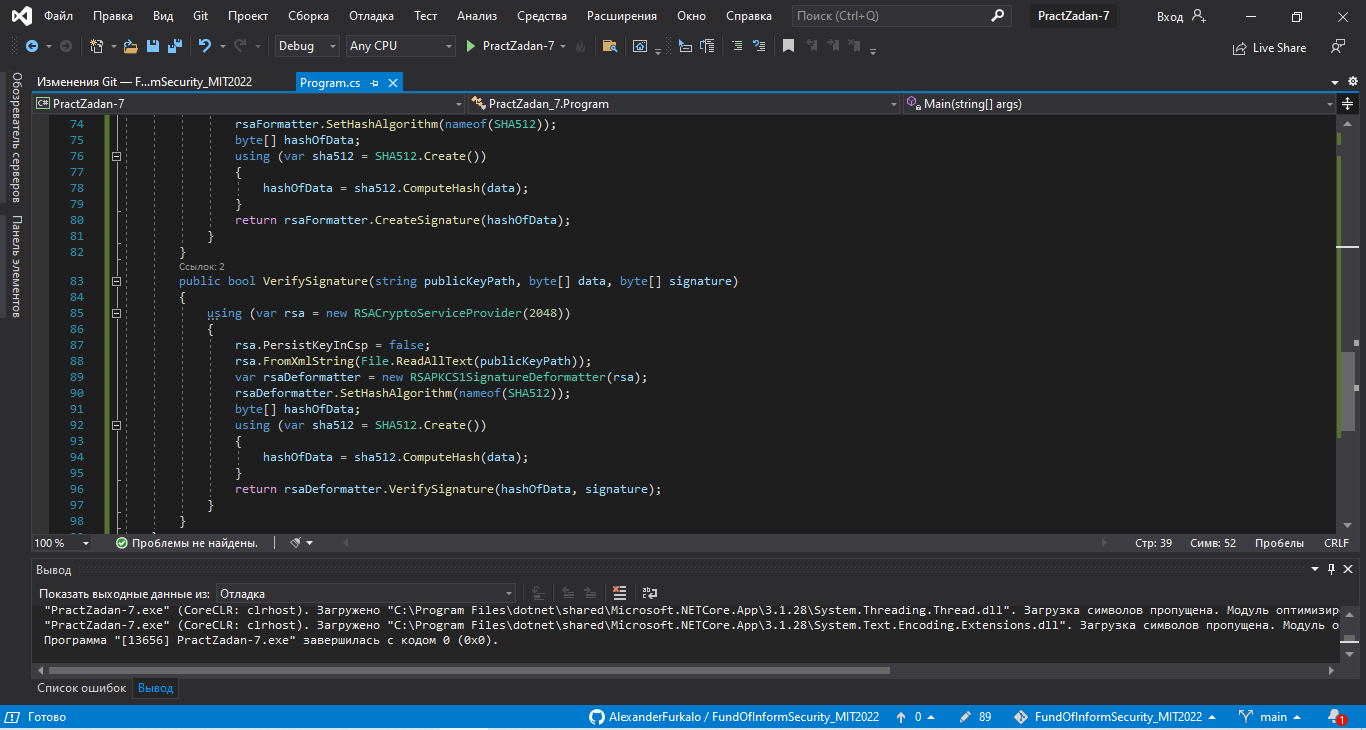
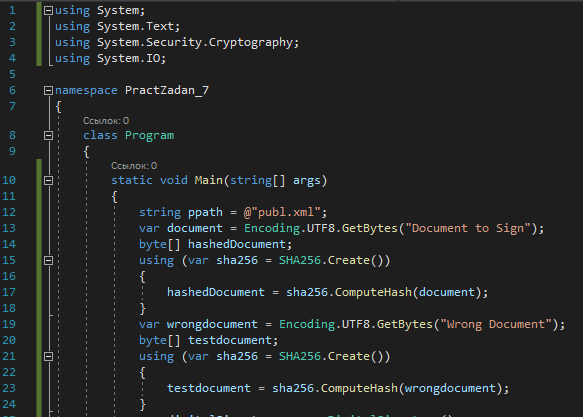
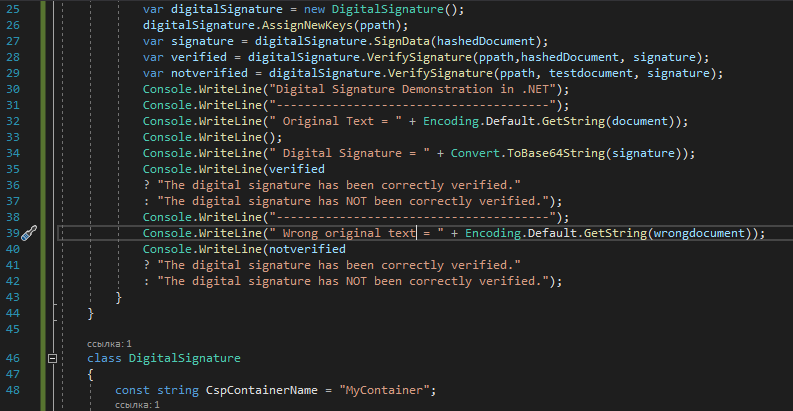
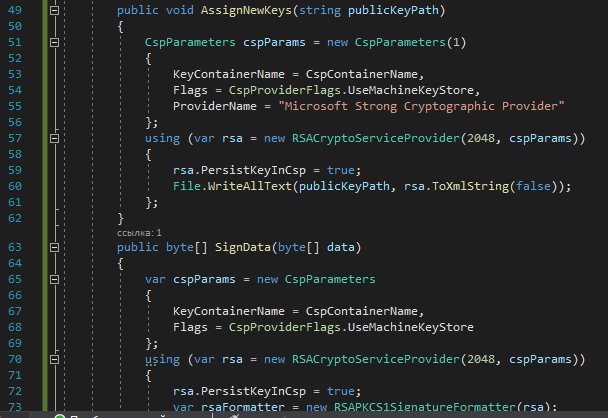
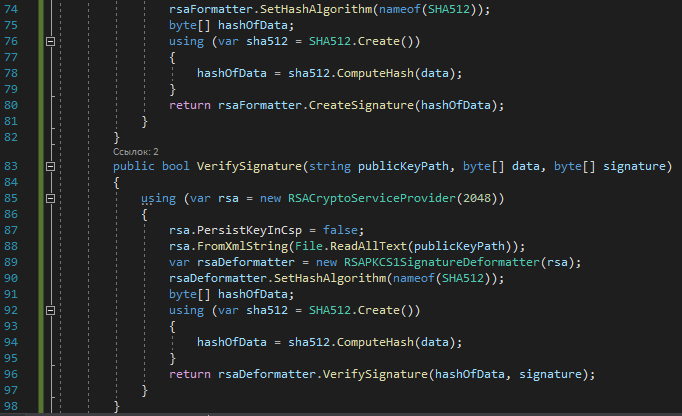
В якості електронного цифрового підпису може виступати сам текст, зашифрований закритим ключем відправника, але такий варіант неефективний (бо шифрування та розшифрування тексту займає набагато більше часу, та тому що довжина ЕЦП буде дорівнювати або перевищувати довжину повідомлення). Сучасні алгоритми ЕЦП базуються на використанні хеш-функцій.

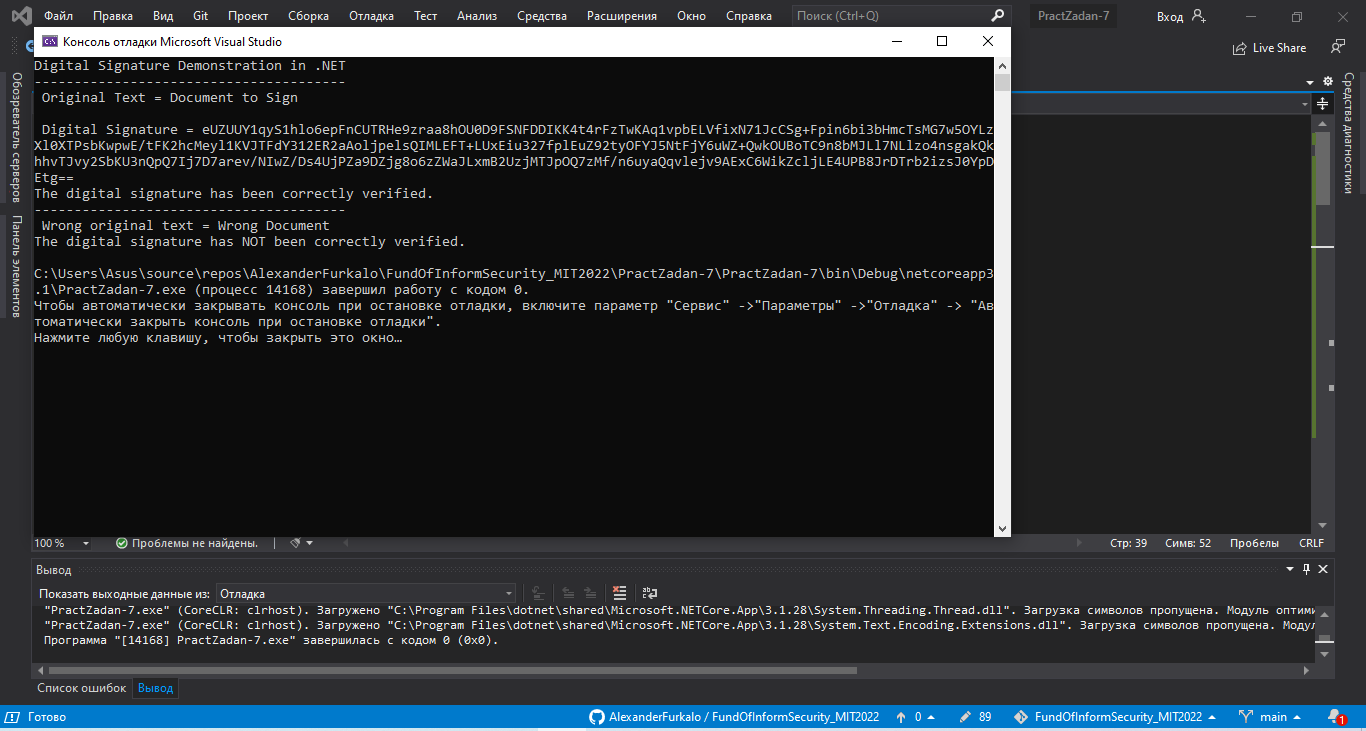
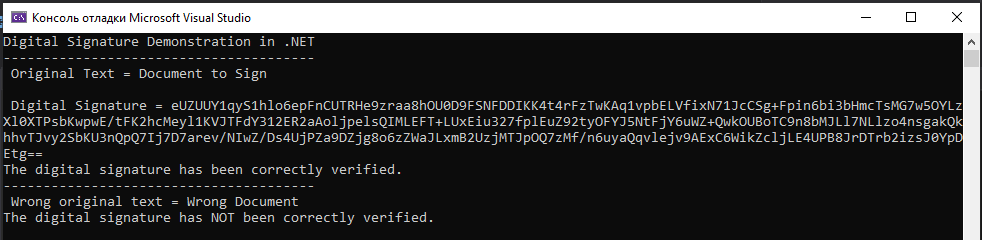
Цифровий підпис складається з наступних трьох алгоритмів:  
- Генерація відкритих і закритих ключів за допомогою RSA;  
- Алгоритм, який використовує закритий ключ для створення підпису;  
- Алгоритм перевірки підпису, який використовує відкритий ключ для перевірки автентичності повідомлення

При підписанні електронного документа до нього додається блок даних (ЕЦП), отримується він за два етапи:   
- Обчислюється (за допомогою програмного забезпечення та математичної функції) “відбиток повідомлення”, який має фіксовану довжину, унікальність та властивість неможливості відтворення повідомлення за лише наявності відбитку  
- Відбиток документа шифрується за допомогою програмного забезпечення (алгоритму ЕЦП) і особистого ключа автора, перетворюючись в ЕЦП

Перевірка займає кілька етапів:  
- Адресат (за допомогою програмного забезпечення сертифікатом відкритого ключа автора) розшифровує підписаний відбиток і одержує відбиток початкового документа;  
- За допомогою програмного забезпечення і спеціальної математичної функції з документа, який був одержаний, обчислюється його відбиток;  
- При перевірці ЕЦП порівнюються відбитки початкового і одержаного документів. Результат – або “вірний”, або “невірний”

Перейдемо до першого завдання:

  
  
  
  
Трохи наблизивши:  
  
  
  


Результати роботи:  
  
  
У цьому завданні ми створили програму, яка підписує повідомлення (“Document to sign”), створює підпис (через SignData) та перевіряє ЕЦП (VerifySignature). Для демонстрації ситуації, коли документ має неправильний підпис, наприкінці Main() є перевірка другого повідомлення (“Wrong document”).  
Спочатку створюється ключі – відкритий ключ зберігається як XML-файл, секретний – у контейнері, використовується метод з попередньої практичної роботи.   
Крім того, потрібно перетворити повідомлення на байтовий масив, і над ним провести операцію хешування (SHA-256).  
Для генерації електронного цифрового підпису документа потрібно для початку ініціювати екземпляр класу CspParameters, що відповідає за параметри криптографічного провайдера (робимо це через var cspParams = new CspParameters). Після цього, ми створюємо екземпляр провайдера асиметричного шифрування RSA (з розміром ключа 2048 та обравши CspParameters), використовуємо секретний ключ з контейнеру та ініціюємо екземпляр класу RSAPKCS1SignatureFormatter (потрібно для підписування). Після цього обчислюємо хеш-код повідомлення (computehash) та обчислюємо ЕЦП (CreateSignature).  
Щоб перевірити підпис, нам потрібно спочатку вказати шлях до відкритого ключа, потрібне повідомлення та ЕЦП, яке будемо перевіряти. У VerifySignature спочатку ініціюємо екземпляр класу RSACryptoServiceProvider (знов з параметром 2048), використовуємо ключ із XML-файлу (у цьому випадку, PersistKeyInCsp вказуємо як false). Майже як для створювання підпису, нам потрібно ініціювати екземпляр класу RSAPKCS1SignatureDeformatter, який потрібен для перевірки. Далі обчислюємо хеш-код повідомлення та перевіряємо ЕЦП (через VerifySignature).

Висновки

У цій практичній роботі, ми вивчили, що таке Електронний Цифровий Підпис (скорочено ЕЦП) та як його використовувати. Також ми написали програму, яка виконує підписування повідомлення та перевірку ЕЦП з використанням алгоритму асиметричного шифрування RSA.