**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Кафедра комп’ютеризованих систем захисту інформації

**КУРСОВА РОБОТА**

**з дисципліни**

**“Прикладна криптологія”**

**Виконав:**

студент БІ-441

Варіант № 2(18)

Безпалий Олександр Русланович

**Перевірив:**

Доцент кафедри КСЗІ Ільєнко А.В.

**Київ 2022**

**Зміст**

**Вступ3**

**Блок-схема7**

**Висновки14**

**Загальний код15**

**Список використаної літератури** **19**

**Вступ**

**Електронний цифровий підпис (ЕЦП)** (англ. *digital signature*) – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

**Надійний засіб електронного цифрового підпису** – засіб електронного цифрового підпису, що має сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації.

Одним з елементів обов'язкового реквізиту є електронний підпис, який використовується для аутентифікації автора та/або підписувача електронного документа іншими суб'єктами електронного документообігу.

Оригіналом електронного документа вважається електронний примірник з електронним цифровим підписом автора.

Електронний цифровий підпис є складовою частиною інфраструктури відкритих ключів.

Електронний цифровий підпис призначений для використання фізичними та юридичними особами — [суб'єктами](https://uk.wikipedia.org/wiki/%D0%A1%D1%83%D0%B1%27%D1%94%D0%BA%D1%82) [електронного документообігу](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%BE%D0%B1%D1%96%D0%B3):

* для [аутентифікації](https://uk.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F) [підписувача](https://uk.wikipedia.org/w/index.php?title=%D0%9F%D1%96%D0%B4%D0%BF%D0%B8%D1%81%D1%83%D0%B2%D0%B0%D1%87&action=edit&redlink=1);
* для підтвердження [цілісності даних](https://uk.wikipedia.org/wiki/%D0%A6%D1%96%D0%BB%D1%96%D1%81%D0%BD%D1%96%D1%81%D1%82%D1%8C_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) в електронній формі.

ЕЦП як спосіб ідентифікації підписувача [електронного документа](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82), дозволяє однозначно визначати походження інформації ([джерело інформації](https://uk.wikipedia.org/wiki/%D0%94%D0%B6%D0%B5%D1%80%D0%B5%D0%BB%D0%BE_%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97)), що міститься у документі. Завдяки цьому ЕЦП є також надійним засобом розмежування відповідальності за інформаційну діяльність у суспільстві, зокрема, відповідальності за [дезінформування](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%B7%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F).

**Юридична вагомість ЕЦП**

Накладання ЕЦП завершує утворення електронного документа, надаючи йому юридичної сили. Згідно з законом України «Про електронні документи та електронний документообіг» юридична сила електронного документа з нанесеними одним або множинними ЕЦП та допустимість такого документа як доказу не може заперечуватися виключно на підставі того, що він має електронну форму.

**Механізм ЕЦП**

Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа. За правовим статусом він прирівнюється до власноручного підпису (печатки). Електронний підпис не може бути визнаний недійсним лише через те, що він має електронну форму або не ґрунтується на посиленому сертифікаті ключа.

За умови правильного зберігання власником секретного (особистого) ключа його підробка неможлива. Електронний документ також не можливо підробити: будь-які зміни, несанкціоновано внесені в текст документа, будуть миттєво виявлені.

**Особистий ключ ЕЦП**

Особистий ключ ЕЦП формується на підставі абсолютно випадкових чисел, що генеруються давачем випадкових чисел, а відкритий ключ обчислюється з особистого ключа ЕЦП так, щоб одержати другий з першого було неможливо.

Особистий ключ ЕЦП є унікальною послідовністю символів довжиною 264 біти, яка призначена для створення Електронного цифрового підпису в електронних документах. Працює особистий ключ тільки в парі з відкритим ключем. Особистий ключ необхідно зберігати в таємниці, адже будь-хто, хто дізнається його, зможе підробити Електронний цифровий підпис.

Документ підписується ЕЦП за допомогою особистого ключа ЕЦП, який існує в одному екземплярі тільки у його власника. Цьому особистому ключу відповідає відкритий ключ, за допомогою якого можна перевірити відповідність ЕЦП його власнику.

**Відкритий ключ ЕЦП і Сертифікат відкритого ключа**

Відкритий ключ використовується для перевірки ЕЦП одержуваних документів (файлів). Відкритий ключ працює тільки в парі з особистим ключем. Відкритий ключ міститься в Сертифікаті відкритого ключа, і підтверджує приналежність відкритого ключа ЕЦП певній особі. Крім самого відкритого ключа, Сертифікат відкритого ключа містить в собі персональну інформацію про його власника (ім'я, реквізити), унікальний реєстраційний номер, термін дії Сертифікату відкритого ключа. З метою забезпечення цілісності представлених у Сертифікаті даних він підписується особистим ключем Центру сертифікації ключів. Сертифікат відкритого ключа може публікуватися на сайті відповідного ЦСК відповідно до Договору про надання послуг ЕЦП.

**Підписання електронного документа ЕЦП**

При підписанні електронного документа його початковий зміст не змінюється, а додається блок даних, так званий «Електронний цифровий підпис». Отримання цього блоку можна розділити на два етапи:

* На першому етапі за допомогою програмного забезпечення і спеціальної математичної функції обчислюється так званий «відбиток повідомлення» (англ. *message digest*).

Цей відбиток має такі властивості:

* фіксовану довжину, незалежно від довжини повідомлення;
* унікальність відбитку для кожного повідомлення;
* неможливість відновлення повідомлення за його відбитком.

Таким чином, якщо документ був модифікований, то зміниться і його відбиток, що відобразиться при перевірці Електронного цифрового підпису.

* На другому етапі відбиток документа шифрується за допомогою програмного забезпечення й особистого ключа автора.

Розшифрувати ЕЦП і одержати початковий відбиток, який відповідатиме документу, можна тільки використовуючи Сертифікат відкритого ключа автора.

Таким чином, обчислення відбитку документа захищає його від модифікації сторонніми особами після підписання, а шифрування особистим ключем автора підтверджує авторство документа.

**Перевірка ЕЦП одержаного документа**

Перевірка Електронного цифрового підпису одержаного документа проводиться декількома етапами:

1. На першому етапі адресат за допомогою програмного забезпечення Сертифікатом відкритого ключа автора розшифровує підписаний відбиток і одержує відбиток початкового документа.
2. За допомогою програмного забезпечення і спеціальної математичної функції з документа, який був одержаний, обчислюється його відбиток.
3. При перевірці ЕЦП порівнюються відбитки початкового і одержаного документів. Результат перевірки — одна з відповідей: «вірний»/«невірний».

**Схеми ЕЦП**

Криптосистема RSA належить до числа перших криптосистем з відкритим ключем з підтримкою електронного цифрового підпису.

У широкому вжитку також знаходяться криптосистеми DSA та ECDSA.

У 2011 році була представлена розширена криптосистема цифрового підпису Меркле (англ. *eXtended Merkle Signature Scheme*, XMSS), яка має такі важливі властивості, як пряма секретність та стійкість до криптоаналізу із використанням квантових комп'ютерів. Заради спрощення впровадження даної криптосистеми були розпочаті роботи над стандартом RFC 8391.

**Використання**

**Властивості інформації**

Електронний цифровий підпис підтверджує достовірність і цілісність документа. Якщо в документ в процесі пересилки були внесені які-небудь зміни, нехай навіть зовсім незначні, то підміна виявиться. Сертифікат відкритого ключа містить персональну інформацію про власника, що дозволяє однозначно ідентифікувати автора документа.

**Фіксація точного часу підписання**

Однією з додаткових можливостей при роботі з ЕЦП є послуга фіксації точного часу підписання документа ЕЦП відмітка точного часу. Відмітка точного часу при підписанні документа дозволяє точно ідентифікувати момент накладання підпису, причому змінити його значення згодом, навіть особою, яка наклала підпис, неможливо. Можливе лише повторне підписання з фіксацією нового часу. Точне значення часу, який використовується для формування відмітки точного часу, здійснюється апаратними засобами Центру сертифікації ключів шляхом синхронізації з джерелами точного часу з точністю до 1 секунди.

**Послуги з надання ЕЦП**

Послуги з надання ЕЦП в Україні впроваджуються акредитованими центрами сертифікації ключів.

Актуальний перелік акредитованих центрів сертифікації ключів публікується на сайті Центрального засвідчувального органу (Акредитовані ЗЦ та ЦСК [Архівовано 8 січня 2012 у Wayback Machine.])

**Блок-схема**

****

Рисунок 1 – блок-схема

**Опис програми**

**Спочатку ініціалізується програма.**

*using System.Numerics;*

*using System.Security.Cryptography;*

*using System.Security.Cryptography.X509Certificates;*

*using System.Security.Policy;*

*public Form1()*

*{*

*InitializeComponent();*

*}*

**На початку ініціалізіції програми додаються бібліотеки, а також запускається головна форма.**

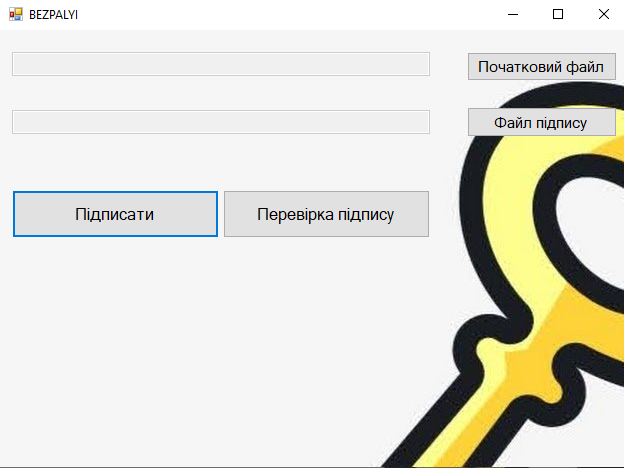


Рисунок 2 – вікно програми

**Далі за логікою програми потрібно обирати файл початковий файл. Для цього потрібно натиснути на кнопку «Початковий файл»(рисунок 3).**

****

Рисунок 3 – Початковий файл

*private void sourceFileButton\_Click(object sender, EventArgs e)*

*{*

*OpenFileDialog ofd = new OpenFileDialog();*

*if (ofd.ShowDialog() == DialogResult.OK)*

*{*

*sourceFilePathTextBox.Text = ofd.FileName;*

*} }*

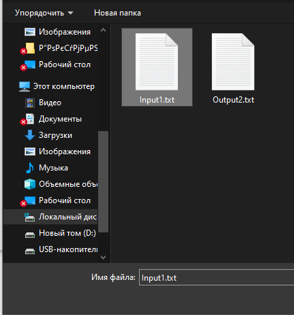


Рисунок 4 – вікно обрання файлу

**Після цього потрібно обрати файл підпису який буде перевірятися. Для цього потрібно натиснути на кнопку «Файл підпису»(рисунок 5).**



Рисунок 5 – вікно програми з обраними файлами

**Для генерації ключа потрібно натиснути кнопку «Підписати»( рисунок 6).**

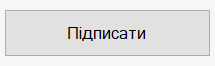


Рисунок 6 – введення простих чисел

*private void buttonEncrypt\_Click(object sender, EventArgs e)*

*{*

*if (sourceFilePathTextBox.Text.Length > 0)*

*{*

*SHA256 alg = SHA256.Create();*

*string message = File.ReadAllText(sourceFilePathTextBox.Text);*

*byte[] data = Encoding.ASCII.GetBytes(message);*

*byte[] hash = alg.ComputeHash(data);*

*RSAParameters sharedParameters;*

*byte[] signedHash;*

*byte[] publicKey;*

*byte[] privateKey;*

*using (RSA rsa = RSA.Create())*

Ключ генеруєтсья завдяки класу *Microsoft using System.Security.Cryptography.* Довжина ключа – 1024, а хеш-функція Sha256. Хеш-функції призначені для створення "відбитків" або "дайджестів" для повідомлень довільної довжини.

Хеш-функції сімейства SHA-2 побудовані на основі структури Меркла-Дамгора. Вихідне повідомлення після доповнення розбивається на блоки, кожен блок – на 16 слів. Алгоритм пропускає кожен блок повідомлення через цикл із 64 або 80 ітераціями (раундами).

На кожній ітерації 2 слова перетворюються, функцію перетворення задають інші слова. Результати обробки кожного блоку складаються, сума є значенням хеш-функції. Тим не менш, ініціалізація внутрішнього стану здійснюється результатом обробки попереднього блоку. Тому незалежно обробляти блоки та складати результати не можна.

*{*

*rsa.KeySize = 1024;*

*sharedParameters = rsa.ExportParameters(true);*

*var keys = rsa.ToXmlString(true);*

*File.WriteAllText("output", keys);*

*RSAPKCS1SignatureFormatter rSAPKCS1SignatureFormatter = new RSAPKCS1SignatureFormatter(rsa);*

*RSAPKCS1SignatureFormatter rsaFormatter = rSAPKCS1SignatureFormatter;*

*rsaFormatter.SetHashAlgorithm(nameof(SHA256));*

*signedHash = rsaFormatter.CreateSignature(hash);*

*File.WriteAllBytes("Signed", signedHash);*

*MessageBox.Show("Файл підписано.");*

*} }*

*else*

*MessageBox.Show("Виберіть файл для шифрування");*

*}*

**Далі програма створить файл-ключ Signed, який не можна буде переглянути(але як приклад всередині нього буде бодібне як на рисунку 7).**

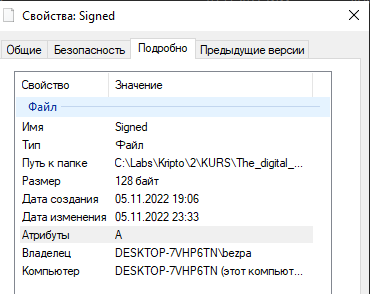


Рисунок 7 – створений ключ

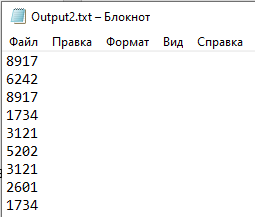


Рисунок 8 – створений ключ

**Для перевірки файлу потрібно натиснути кнопку «Перевірка підпису»(рисунок 9). Якщо все вірно, то обраний файл буде перевірено і буде отримано результат.**

*private void buttonDecipher\_Click(object sender, EventArgs e)*

*{*

*if ((sourceFilePathTextBox.Text.Length > 0) && (signFilePathTextBox.Text.Length > 0))*

*{*

*SHA256 alg = SHA256.Create();*

*byte[] signedHash = File.ReadAllBytes(signFilePathTextBox.Text);*

*byte[] data = File.ReadAllBytes(sourceFilePathTextBox.Text);*

*byte[] hash = alg.ComputeHash(data);*

*using (RSA rsa = RSA.Create())*

*{*

*rsa.FromXmlString(File.ReadAllText("output"));*

*RSAPKCS1SignatureDeformatter rsaDeformatter = new RSAPKCS1SignatureDeformatter(rsa);*

*rsaDeformatter.SetHashAlgorithm(nameof(SHA256));*

*if (rsaDeformatter.VerifySignature(hash, signedHash))*

*{*

*MessageBox.Show("Файл підписано. Підпис вірний.");*

*}*

*else*

*{*

*MessageBox.Show("Увага! Файл НЕ пройшов перевірку!!!");*

*}*

*}*

*}*

*else*

*MessageBox.Show("Виберіть файл для дешифрування та/або файл підпису");*

*}*

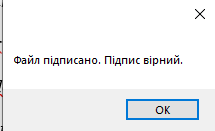


Рисунок 9 – результат перевірки

**Якщо ж з якоїсь причини файл не пройшов перевірку то буде виведено помилку (рисунок 10).**

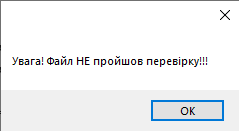


Рисунок 10 – створений ключ

**Висновки**

**Електронний цифровий підпис** (або скорочено – ЕЦП) за правовим статусом прирівняний до власноручного підпису або печатки.

ЕЦП – це дані в електронній формі, отримані за результатами криптографічного перетворення, які додаються до інших даних або документів і забезпечують їх цілісність та ідентифікацію автора.

За допомогою послуг ЕЦП можна підписувати електронні документи, користуватися електронними послугами, реєструватися на державних порталах тощо. Документи, підписані за допомогою ЕЦП, мають таку саму юридичну силу, як і звичайні.

Станом на середину 2018 року близько 9 мільйонів фізичних осіб та представників юридичних осіб та вже мають ЕЦП, серед них третина – фізичні особи, фізичні особи-підприємці та самозайняті особи.

Отримати послуги ЕЦП фізична або юридична особа може в одному з Акредитованих центрів сертифікації ключів (АЦСК), повний перелік яких наведено в Електронному реєстрі суб’єктів, які надають послуги, пов’язані з ЕЦП

***System.Security.Cryptography Namespace***

Надає криптографічні послуги, включаючи безпечне кодування та декодування даних, а також багато інших операцій, таких як хешування, генерація випадкових чисел і автентифікація повідомлень. Щоб отримати додаткові відомості, див. Криптографічні служби.

**Загальний код**

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Diagnostics;*

*using System.Drawing;*

*using System.IO;*

*using System.Linq;*

*using System.Text;*

*using System.Threading.Tasks;*

*using System.Windows.Forms;*

*using System.Numerics;*

*using System.Security.Cryptography;*

*using System.Security.Cryptography.X509Certificates;*

*using System.Security.Policy;*

*namespace protect\_inf\_LR1*

*{*

*public partial class Form1 : Form*

*{*

*public Form1()*

*{*

*InitializeComponent();*

*}*

*//ШИФРОВКА*

*private void buttonEncrypt\_Click(object sender, EventArgs e)*

*{*

*if (sourceFilePathTextBox.Text.Length > 0)*

*{*

*SHA256 alg = SHA256.Create();*

*string message = File.ReadAllText(sourceFilePathTextBox.Text);*

*byte[] data = Encoding.ASCII.GetBytes(message);*

*byte[] hash = alg.ComputeHash(data);*

*RSAParameters sharedParameters;*

*byte[] signedHash;*

*byte[] publicKey;*

*byte[] privateKey;*

*using (RSA rsa = RSA.Create())*

*{*

*rsa.KeySize = 1024;*

*sharedParameters = rsa.ExportParameters(true);*

*var keys = rsa.ToXmlString(true);*

*File.WriteAllText("output", keys);*

*RSAPKCS1SignatureFormatter rSAPKCS1SignatureFormatter = new RSAPKCS1SignatureFormatter(rsa);*

*RSAPKCS1SignatureFormatter rsaFormatter = rSAPKCS1SignatureFormatter;*

*rsaFormatter.SetHashAlgorithm(nameof(SHA256));*

*signedHash = rsaFormatter.CreateSignature(hash);*

*File.WriteAllBytes("Signed", signedHash);*

*MessageBox.Show("Файл підписано.");*

*}*

*}*

*else*

*MessageBox.Show("Виберіть файл для шифрування");*

*}*

*//ДЕШИФРОВКА*

*private void buttonDecipher\_Click(object sender, EventArgs e)*

*{*

*if ((sourceFilePathTextBox.Text.Length > 0) && (signFilePathTextBox.Text.Length > 0))*

*{*

*SHA256 alg = SHA256.Create();*

*byte[] signedHash = File.ReadAllBytes(signFilePathTextBox.Text);*

*byte[] data = File.ReadAllBytes(sourceFilePathTextBox.Text);*

*byte[] hash = alg.ComputeHash(data);*

*using (RSA rsa = RSA.Create())*

*{*

*rsa.FromXmlString(File.ReadAllText("output"));*

*RSAPKCS1SignatureDeformatter rsaDeformatter = new RSAPKCS1SignatureDeformatter(rsa);*

*rsaDeformatter.SetHashAlgorithm(nameof(SHA256));*

*if (rsaDeformatter.VerifySignature(hash, signedHash))*

*{*

*MessageBox.Show("Файл підписано. Підпис вірний.");*

*}*

*else*

*{*

*MessageBox.Show("Увага! Файл НЕ пройшов перевірку!!!");*

*}*

*}*

*}*

*else*

*MessageBox.Show("Виберіть файл для дешифрування та/або файл підпису");*

*}*

*private void sourceFileButton\_Click(object sender, EventArgs e)*

*{*

*OpenFileDialog ofd = new OpenFileDialog();*

*if (ofd.ShowDialog() == DialogResult.OK)*

*{*

*sourceFilePathTextBox.Text = ofd.FileName;*

*}*

*}*

*private void signFileButton\_Click(object sender, EventArgs e)*

*{*

*OpenFileDialog ofd = new OpenFileDialog();*

*if (ofd.ShowDialog() == DialogResult.OK)*

*{*

*signFilePathTextBox.Text = ofd.FileName;*

*}*

*}*

*private void Form1\_Load(object sender, EventArgs e)*

*{*

*}*

*private void textBox\_p\_TextChanged(object sender, EventArgs e)*

*{*

*}*

*}*

*}*

**Список використаної літератури**

1. [**https://czo.gov.ua/sign**](https://czo.gov.ua/sign)
2. [**https://www.kmu.gov.ua/usi-pitannya-po-e-poslugam/sho-tak-elektronnij-cifrovij-pidpis-ecp**](https://www.kmu.gov.ua/usi-pitannya-po-e-poslugam/sho-tak-elektronnij-cifrovij-pidpis-ecp)
3. [**https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9\_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B8%D0%B9\_%D0%BF%D1%96%D0%B4%D0%BF%D0%B8%D1%81**](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D1%96%D0%B4%D0%BF%D0%B8%D1%81)
4. [**https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography?view=net-6.0**](https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography?view=net-6.0)