МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ С.В. Казмірчук
«____» _____ 2021 р.

На правах рукопису
УДК 004.056:004.02(079.2)

ДИПЛОМНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

Тема: Метод захисту даних з використанням електронного цифрового підпису

Виконавець: А.Я. Сарапіна
Керівник: старший викладач О.Л. Яковенко
Нормоконтролер: старший викладач О.Л. Яковенко

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки, комп'ютерної та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: «Бакалавр»

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ С.В. Казмірчук

«_____»_____ 2021 р.

ЗАВДАННЯ

на виконання дипломної роботи

здобувача вищої освіти Сарапіної Альони Ярославівни

1. Тема: *Метод захисту даних з використанням електронного цифрового підпису* затверджена наказом ректора від «26» квітня 2021 р. № 652/ст.
2. Термін виконання: з 10.05.2021 р. по 20.06.2021 р.
3. Вихідні дані: провести аналіз існуючих методів захисту даних; здійснити аналіз алгоритмів реалізації електронного цифрового підпису; на основі проведеного аналізу обрати алгоритм для подальшої реалізації; здійснити програмну реалізацію обраного алгоритму підпису;
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): аналіз існуючих методів захисту даних; аналіз принципів роботи ЕЦП, його значення у інформаційній безпеці; аналіз та порівняння алгоритмів цифрового підпису; програмна реалізація DSA алгоритму; створення програмного модулю методу захисту даних з використанням реалізованого алгоритму; тестування створеного методу; аналіз отриманих результатів.

КАЛЕНДАРНИЙ ПЛАН

виконання дипломної роботи

№ з/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Уточнення постановки задачі	19.04.2021	Виконано
2	Аналіз літературних джерел	20.04.2021	Виконано
3	Обґрунтування рішення	23.04.2021	Виконано
4	Збір інформації	26.04.2021	Виконано
5	Аналіз методів захисту даних	02.05.2021	Виконано
6	Аналіз існуючих алгоритмів ЕЦП	10.05.2021	Виконано
7	Програмна реалізація обраного алгоритму	17.05.2021	Виконано
8	Аналіз створеного методу захисту даних	24.05.2021	Виконано
9	Перевірка на антиплагіат	29.05.2021	Виконано
10	Оформлення і друк пояснювальної записки	30.05.2021	Виконано
11	Оформлення презентації	02.06.2021	Виконано
12	Отримання рецензій від рецензентів	10.06.2021	Виконано

Здобувач вищої освіти А. Я. Сарапіна

(підпис, дата)

Керівник дипломної роботи О. Л. Яковенко

(підпис, дата)

РЕФЕРАТ

Дипломна робота включає вступ, основну частину, яка складається з двох розділів, загальних висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 58 сторінок, які містять 28 рисунків, 2 таблиці, 7 сторінок додатків. Список використаних джерел містить 19 найменувань та займає 2 сторінки від загального обсягу роботи.

Метою роботи є створення методу захисту даних з використанням електронного цифрового підпису.

В дипломній роботі вирішено задачу побудови авторського програмного модулю алгоритму електронного цифрового підпису.

В роботі розроблено DSA алгоритм підпису та створено програмний модуль для генерації ключів, створення ЕЦП та його верифікації.

Створений метод захисту даних та програмний модуль стосуються галузі безпеки інформаційних систем і можуть використовуватись для забезпечення цілісності та конфіденційності.

Ключові слова: методи захисту даних, криптографічні методи, електронний цифровий підпис, алгоритми підпису, Digital Signature Algorithm (DSA), відкритий ключ, особистий ключ, підпис документу, верифікація документу.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ЕЦП – електронний цифровий підпис

ЕП – електронний підпис

RSA – абревіатура від прізвищ Rivest, Shamir та Adleman; криптографічний алгоритм

DSA – Digital Signature Algorithm

ECDSA – Elliptic Curve Digital Signature Algorithm

ПФЧ – проблема факторизації цілих чисел

ПДЛ – проблема дискретного логарифму

ПДЛЕК – проблема дискретного логарифму в групі точок еліптичної кривої

MD – Message Digest Algorithm

SHA – Secure Hash Algorithm

VS – Visual Studio

JS – JavaScript

SPKI – Simple public-key infrastructure

ВСТУП

Актуальність теми. В зв’язку з постійним розвитку інформаційних технологій створюються додаткові вимоги до захисту персональних даних. Існує нагальна загроза цілісності та конфіденційності інформації, тому захист важливих даних повинен бути ретельно продуманий. Двадцять років тому проблема захисту інформації вирішувалась за допомогою криптографічного шифрування, встановлення брандмауера та обмеження доступу. В даний час цих технологій недостатньо, і будь-яка інформація, що має фінансову, конкурентну, військову чи політичну цінність, знаходиться під загрозою. Іншою загрозою є можливість перехоплення управління критично важливою інформаційною інфраструктурою.

Найважливішою складовою частиною дотримання політики ІБ практично на всіх підприємствах є належний рівень забезпечення безпеки персональних даних. Політика держави спрямована на посилення заходів контролю в цій сфері. Згідно з Законом «Про захист персональних даних» : законодавство про захист персональних даних складають Конституція України, цей Закон, інші закони та підзаконні нормативно-правові акти, міжнародні договори України, згода на обов'язковість яких надана Верховною Радою України [1].

Метою роботи є створення методу захисту даних з використанням електронного цифрового підпису.

Для досягнення поставленої мети роботи необхідно вирішити такі задачі:

- провести аналіз існуючих методів захисту даних
- дослідити методи захисту даних з використанням електронного цифрового підпису
- розробити програмний модуль електронного цифрового підпису

Об’єкт дослідження – процес захисту даних з використанням електронного цифрового підпису

Предмет дослідження – методи захисту даних на основі створеного електронного цифрового підпису

Галузь застосування. Результати здійснених у дипломній роботі досліджень можуть бути використані та впроваджені суб’єктами інформаційної безпеки під час захисту цілісності даних.

Практична значимість. Полягає в розробці програмного криптографічного алгоритму DSA з використанням інтегрованого середовища розробки для мов програмування C/C++ та редактора вихідних кодів Visual Studio Code, що дало змогу забезпечити цілісність інформаційних ресурсів.

РОЗДІЛ 1. МЕТОДИ ЗАХИСТУ ДАНИХ

Інформація відіграє провідну роль у забезпеченні безпеки всіх об’єктів у суспільстві. Це пояснює той факт, що запобігання витоку інформації є найважливішою діяльністю країни.

Вся існуюча інформація існує на різних фізичних носіях і у різних формах:

- письмовій формі;
- усній або слуховій формі
- телекомунікаційній формі.

Інформація про документ міститься на паперових та магнітних носіях у графічному та буквено-цифровому вигляді. Головною її особливістю є те, що вона містить дані, які потрібно захистити у стислій формі. Голосова інформація формується під час переговорного процесу та під час роботи системи відтворення або посилення звуку. Носієм такої інформації можуть бути акустичні механічні коливання, що поширюються від джерела до космічного простору. Інформація про телекомунікації походить від технічних засобів зберігання та обробки даних під час передачі по каналам зв'язку. У цьому випадку носієм інформації є електричний струм. І якщо дані передаються через оптичний канал і радіоканал, несучою хвилею є електромагнітна хвиля.

1.1 Основні методи захисту даних в залежності від об'єкта

Основними об'єктами інформації можуть стати:

- Інформаційні ресурси. Ці ресурси можуть містити інформацію, що стосується державної таємниці та конфіденційних даних.

- Засоби та системи інформатизації (системи та мережі, інформаційні та комп’ютерні системи), програмне забезпечення (операційні системи, СУБД та інші типи програм та системного програмного забезпечення), системи зв’язку. Тобто сюди входять інструменти та системи, які обробляють лише «закриту» інформацію. Ці системи та засоби вважаються технічними засобами обробки, прийому, передачі та зберігання даних.
- Технічні методи і системи, розташовані в місці, де обробляються конфіденційні дані. Ці технічні засоби та системи є допоміжними [2].

Зважаючи на це найбільш поширеними методами захисту даних є:

Перешкодження. Це метод захисту інформації, який передбачає фізичне обмеження шляху до захищеного носія.

Управління доступом. Це означає захист інформаційних ресурсів шляхом контролювання використанням кожного ресурсу. Ці методи включають апаратне та програмне забезпечення та елементи бази даних.

Маскування. Це метод захисту інформації, що означає криптографічне закриття даних. При передачі даних по довгих каналах цей метод вважається єдиним надійним методом.

Регламентація. Цей метод передбачає захист інформації та даних, тим самим мінімізуючи можливість несанкціонованого доступу.

Примус. Це метод захисту інформації, під час якого персонал та користувачі системи повинні дотримуватися правил поводження з усіма захищеними даними – передача, обробка та використання інформації. Якщо вони не відповідають цим умовам, можливо, їм доведеться нести адміністративну або матеріальну відповідальність.

Спонукання. Сюди відносяться такі методи захисту інформації, при яких персонал і користувача спонукають дотримуватися встановленого порядку, дотримуватись етичних та моральних норм (прописаних і регламентованих).

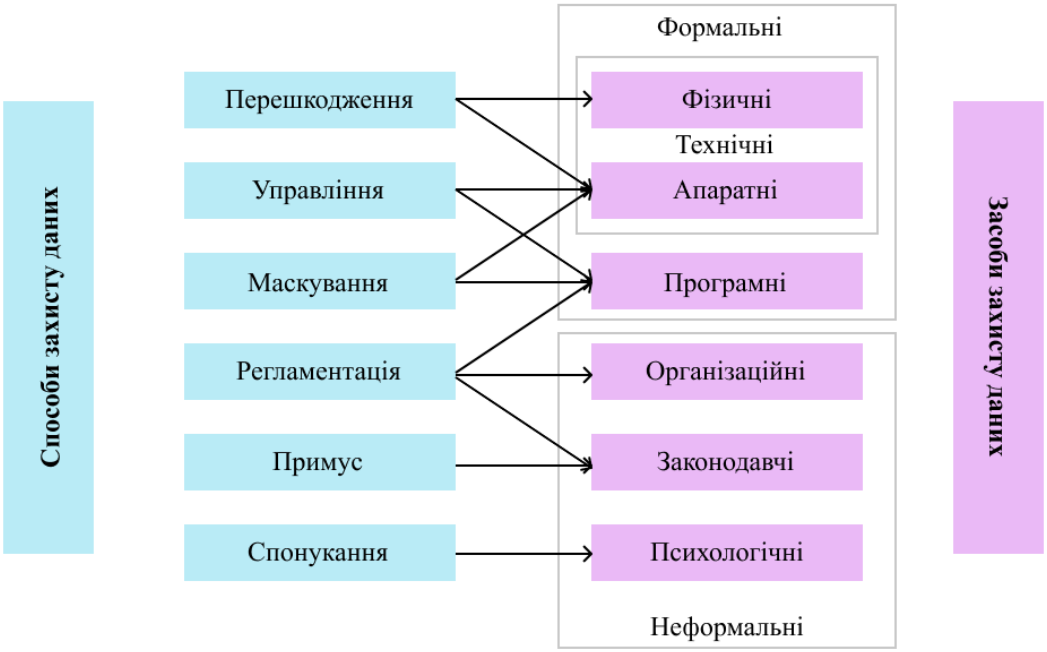


Рис 1. Основні методи захисту даних

1.1.1 Методи захисту даних, які обмежують доступ

Ідентифікація ресурсів комп'ютерної мережі, їх користувачів і персоналу (присвоєння об'єктам персонального ідентифікатора).

Метод пізнання або дійсності об'єкта по тому ідентифікатором, який був зазначений при вході в систему.

Контроль повноважень, який має на увазі аналіз відповідності часу доби, дня, ресурсів і запитуваних процедур згідно із встановленим регламентом.

Встановлення регламенту для того, щоб дозволити діапазон робочого часу.

Реєстрація кожного звернення до тих ресурсів, що захищаються.

Реакція обмеження в разі вчинення спроби несанкціонованого доступу (відмова в запиті або включення сигналізації).

Існуючі методи захисту інформації дуже різноманітні, та потребують великої уваги, але їх однозначно необхідно використовувати у всіх сферах повсякденного життя.

1.1.2 Організаційні методи захисту даних

До компетенції служби безпеки обов'язково повинна входити розробка комплексу організаційних засобів захисту інформації. Найчастіше компетентні фахівці застосовують такі методи інформаційного захисту:

Розробляють внутрішні нормативні документи, в яких повинні бути встановлені правила роботи з конфіденційною інформацією і комп'ютерною технікою.

Проводять періодичні перевірки персоналу і інструктажі стосовно збереження конфіденційних даних. Крім цього повинно ініціюватись підписання додаткових угод до трудового договору, в яких чітко прописана відповідальність працівника за неправомірне використання або розголошення відомостей, які стали йому відомі в процесі здійснення його професійної діяльності.

Служба безпеки також повинна розмежувати зони відповідальності для виключення тих ситуацій, коли найбільш важлива інформація знаходиться в доступі тільки одного співробітника. Крім перерахованих вище методів захисту інформації компетентні співробітники повинні організувати роботу в загальних програмах документообігу і простежити, щоб особливо важливі файли не зберігалися поза мережевих дисків.

Впроваджують програмні комплекси, які захищають інформацію від знищення або копіювання будь-яким користувачем системи, в тому числі топ-менеджером компанії.

Складають плани, які можуть відновити систему в тому випадку, якщо вона вийде з ладу.

1.1.3 Технічні методи захисту даних

Основні технічні методи захисту інформації включають програмні і апаратні засоби. До них можна віднести:

Забезпечення віддаленого зберігання і резервного копіювання найбільш важливих інформаційних даних на регулярній основі.

Резервування і дублювання всіх підсистем, які містять важливу інформацію.

Перерозподіл ресурсів мережі в тому випадку, якщо порушена працездатність її окремих елементів.

Забезпечення можливості застосовувати резервні системи електричного живлення.

Забезпечення безпеки інформаційних даних та належного захисту в разі виникнення пожежі або пошкодження комп'ютерного обладнання водою.

Установка такого програмного забезпечення, яке зможе забезпечити належний захист інформаційних баз даних у разі несанкціонованого доступу.

1.1.4 Криптографічні методи захисту даних

Метою застосування криптографічних методів є захист інформаційної системи від цілеспрямованих руйнівних впливів (атак) з боку зловмисника[3].

Основні завдання криптографії [4]:

- Забезпечення конфіденційності даних (запобігання несанкціонованого доступу до даних). Це одне з основних завдань криптографії, для його вирішення використовують шифрування даних, тобто дані перетворюють так, щоб прочитати їх могли лише

користувачі, які мають законе право на доступ до них, і володіють відповідним ключем.

- Забезпечення цілісності даних - гарантії того, що при передачі або зберіганні дані не були змінені користувачами, які не мають на це прав. Під модифікацією мається на увазі редагування, видалення або заміна даних, а також повторне пересилання тексту, який був перехоплений раніше.
- Забезпечення аутентифікації. Під аутентифікацією розуміється перевірка суб'єктів на реальність або даних на правдивість. У більшості випадках суб'єкт А повинен не просто довести свої права, а зробити це так, щоб перевіряючий суб'єкт (В) не зміг згодом сам користуватися отриманою інформацією з метою видати себе за суб’єкта А. Такі докази називаються «доказами з нульовим розголошенням».
- Забезпечення неможливості відмови від авторства - запобігання можливості відмови суб’єктів від їх дій (зазвичай - неможливість відмовити у підписанні документа). Це завдання схоже з протилежним - гарантування того, що авторство неможливо приписати комусь іншому. Найяскравішим прикладом ситуації, коли виникає таке завдання, є підписання контракту двома або більше людьми, які не довіряють один одному. У цій ситуації всі договірні сторони повинні бути впевнені, що в майбутньому жодна з них не зможе відмовитись від підписання, та ніхто не зможе змінити, замінити або створити новий документ (угоду) та стверджувати, що цей новий документ було підписано. Основним способом вирішення даної проблеми є використання цифрового підпису.

1.2 Електронний цифровий підпис

Електронний цифровий підпис (ЕЦП) — 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 із 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 [5].

ЕЦП використовується суб’єктами документообігу, а саме фізичними та юридичними особами. Його застосовують для:

1. Аутентифікації людини, що підписує
2. Для того щоб підтвердити цілісність даних в електронному вигляді

За допомогою ЕЦП можна визначити походження (джерело) інформації, яка міститься в документі. Зважаючи на це ЕЦП є надійним способом розмежування інформаційної відповідальності та відповідальності за дезінформування.

Накладання електронного цифрового підпису на документ, надає йому юридичної сили. Згідно закону 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 формух [6].

Електронний цифровий підпис у ролі способу контролю за походженням та цілісністю інформації є ключовим інструментом безпеки інформації на кожному рівні соціальної інфраструктури: починаючи з захисту персональних даних до державної безпеки інформації. Отже, ЕЦП зокрема та інфраструктура відкритих ключів загалом є стратегічними оборонними технологіями, якість та надійність яких залежить від інформаційної безпеки України.

ЕЦП встановлюється із закритим ключем та перевіряється за допомогою відкритого ключа. З боку юридичного статусу у відповідність можна поставити власноручний підпис (печатку). Електронний підпис не можна відкликати просто тому, що він має електронний формат або не ґрунтується на розширеному сертифікаті ключа.

Якщо власник правильно зберігає секретний (приватний) ключ, його не можна підробити. Також неможливо сфальсифікувати електронний документ: будь-які зміни тексту документа без дозволу будуть негайно виявлені.

Приватний ключ ЕЦП генерується на основі повністю випадкових чисел, сформованих датчиком випадкових чисел, а відкритий ключ обчислюється з приватного ключа ЕЦП, так що другий не може бути отриманий з першого.

Закритий ключ ЕЦП - це унікальна 264-бітна послідовність символів, призначена для створення електронного цифрового підпису в електронних документах. Приватний ключ працює лише в парі з відкритим ключем. Приватний ключ потрібно тримати в таємниці, оскільки кожен, хто його дізнається, зможе підробити електронний цифровий підпис.

Документ підписується ЕЦП приватним ключем ЕЦП, який в одному примірнику існує лише для його власника. Цей закритий ключ відповідає відкритому ключу, за допомогою якого ви можете перевірити відповідність ЕЦП його власнику.

Переваги застосування ЕЦП:

- значно зменшує час, витрачений на транзакцію та обмін документацією;
- вдосконалення та зменшення витрат на процес підготовки, доставки, обліку та зберігання документів;
- забезпечує достовірність документації;
- зменшує ризик фінансових втрат за рахунок збільшення конфіденційності обміну інформацією;
- допомагає побудувати систему обміну документами компанії.

1.2.1 Види електронного цифрового підпису

Виділяють три види ЕЦП:

- простий електронний цифровий підпис;
- вдосконалений некваліфікований електронний цифровий підпис;
- вдосконалений кваліфікований електронний цифровий підпис [7].

Простий електронний цифровий підпис

- Використовуючи коди, паролі чи інші засоби, простий електронний цифровий підпис підтверджує той факт, що певна особа створила електронний підпис.

Простий електронний цифровий підпис має низький рівень захисту. Це дозволяє лише вказати автора документа.

Простий електронний цифровий підпис не захищає документ від підробки.

Вдосконалений некваліфікований електронний цифровий підпис

- отриманий в результаті криптографічного перетворення інформації за допомогою ключа електронного підпису;
- надає можливість ідентифікації особи, яка підписала електронний документ;
- дає змогу розкрити факти про зміни в електронному документі після моменту його підписання;
- створюється за допомогою електронних підписів.

Вдосконалений некваліфікований електронний цифровий підпис має середній рівень захисту.

Для використання некваліфікованого електронного підпису вам потрібен сертифікат його ключа підтвердження.

Вдосконалений кваліфікований електронний цифровий підпис

Кваліфікований електронний підпис можна охарактеризувати некваліфікованим електронним підписом.

Вдосконалений кваліфікований електронний цифровий підпис виконує наступні додаткові функції підпису:

- ключ для перевірки електронного підпису вказується у кваліфікованому сертифікаті;
- засоби електронного підпису використовуються для створення та перевірки електронних підписів, які отримали підтвердження відповідності вимогам законодавства.

Вдосконалений кваліфікований електронний цифровий підпис - це найбільш універсальний та стандартизований підпис з високим рівнем захисту.

Документ, підписаний таким підписом, схожий на паперовий варіант із власноручним підписом.

Такий підпис можна використовувати без будь-яких додаткових домовленостей та положень між учасниками електронного документообігу.

Якщо документ має кваліфікований підпис, ви можете точно вказати, який працівник організації його дав.

А також з’ясувати, чи змінився документ після підписання.

1.2.2 Застосування різних видів електронного цифрового підпису

Простий ЕЦП

Юридичні особи подають заявки на державні та комунальні послуги, підписуючи заявку простим ЕП уповноваженої особи.

Використання простого ЕП для отримання державних чи муніципальних послуг дозволяється, якщо федеральні закони чи інші нормативні акти не забороняють подання заявки в державну чи муніципальну службу в електронній формі та не передбачають використання для цього іншого виду електронного підпису.

Вдосконалений некваліфікований ЕЦП

Випадки, коли інформація в електронній формі, підписана некваліфікованим електронним підписом, визнається електронним документом, еквівалентним документу на папері, підписаним власноручним підписом.

Міністерство фінансів вважає, що для цілей податкового обліку документ, виготовлений в електронній формі та підписаний некваліфікованим електронним підписом, не може бути документом, еквівалентним паперовому документу, підписаному власноручним підписом.

Таким чином, хоча обидві сторони можуть, за наявності юридично чинної угоди, організувати електронне управління документами з використанням вдосконаленого некваліфікованого електронного підпису, якщо існує

можливість виникнення суперечок з контролюючим органом, значення таких документів втрачається.

Вдосконалений кваліфікований ЕЦП

Електронну накладну слід підписувати лише посиленим кваліфікованим електронним підписом керівника або інших осіб, уповноважених наказом (іншим розпорядчим документом) або уповноваженням від імені організації, індивідуального підприємця.

Заява про реєстрацію (зняття з реєстрації) в податковому органі підтверджується лише вдосконаленим кваліфікованим підписом [8].

Претензії щодо повернення або заліку суми податку також приймаються, лише якщо вони підтверджують вдосконалений кваліфікований електронний підпис.

1.2.3 Електронний цифровий підпис як метод захисту даних

Електронні документи все частіше з’являються в діловому середовищі та уряді, витісняючи паперові документи. Сьогодні в умовах глобалізації зростаючий обсяг інформації, збільшення обміну інформацією, загрози безпеці електронних документів стають дедалі серйознішими. Розвиток Інтернету, XML, мобільних та бездротових технологій, стандартизація форматів даних та протоколів для їх обміну роблять інформаційне середовище організацій дедалі відкритішою та вразливішою до різних атак зловмисників. Як наслідок, технології захисту електронних систем документообігу від зовнішніх та внутрішніх загроз стають все більш важливими. Тому що неможливо використовувати такі традиційні методи захисту, як електронні печатки або власноручні підписи електронних документів. Натомість використовується електронний цифровий підпис. Згідно з українським законом "Про електронний цифровий підпис", юридичний статус електронного цифрового підпису такий самий, як рукописний підпис та / або печатка, та зазначені умови його визнання [6]. Однак слід зазначити, що використання електронного цифрового підпису

не змінює процедуру підписання договорів та інших документів, передбачених законодавством для письмових операцій.

На рисунку представлена схема електронного цифрового підпису, і там ми бачимо, що електронний цифровий підпис складається із секретного ключа та сертифіката відкритого ключа, що містять відкритий ключ перевірки та підтверджують належність відкритого ключа електронного цифрового підпису конкретній особі . При перевірці ЕЦП порівнюються дані первинного та отриманого документів. Результат тесту є однією з відповідей: "true" / "false".

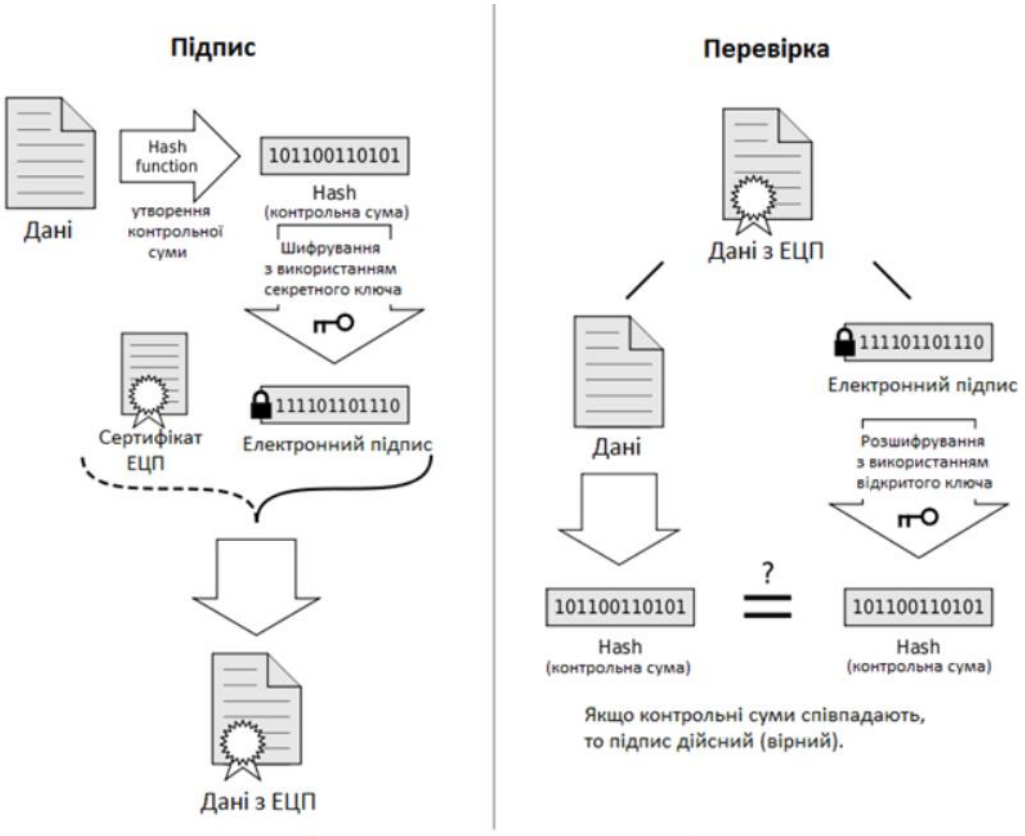


Рис 2. Схема роботи ЕЦП

Електронний цифровий підпис підтверджує справжність та цілісність документа. Якщо під час передачі в документ були внесені будь-які зміни (навіть незначні), відбудеться заміна. Сертифікат відкритого ключа містить особисту інформацію про власника, що дозволяє однозначно ідентифікувати автора документа.

Таким чином, можна зробити висновок, що електронний цифровий підпис є невід'ємною частиною сучасного документообігу, оскільки його використання підвищує рівень захисту даних і спрощує різні процедури (укладання угод, звітування перед державними органами), значно економить час, забезпечує достовірність створеного довору та забезпечує цілісність секретної інформації про конкретну особу.

1.3 Вибір алгоритму підпису

Практичні схеми реалізації ЕЦП:

- Схема RSA
- Схема Ель-Гамаля
- DSA
- ECDSA

1.3.1 Схема RSA

RSA— криптографічний алгоритм з відкритим ключем, що базується на обчислювальній складності задачі факторизації великих цілих чисел [9].

RSA – перший алгоритмом придатним для шифрування і ЕЦП. Він використовується для великої кількості криптографічних додатків.

Генерація ключової пари.

1) Обираються два великих простих числа p і q

2) Обраховується n=p * q.

3) Обирається будь-яке натуральне число

$$e\ (1 < e < \varphi(n)),\ НОД\ (e, \varphi(n)) = 1$$

4) За алгоритмом Евкліда вирішується рівняння з цілими числами

$$e * d + (p-1) * (q-1) * y = 1.$$

5)Вирішивши рівняння, одержуємо значення закритого ключа d.

Формування підпису.

$$s = M^d mod\ n$$

$$(H\ (M),\ s)$$

Перевірка (верифікація) підпису.

$$M' = s^e mod\ n.$$

Якщо $h\ (M)\ =\ hash\ (M')$ - ЕЦП вірний,

$h\ (M)\ \neq\ hash\ (M')$ - ЕЦП не вірний

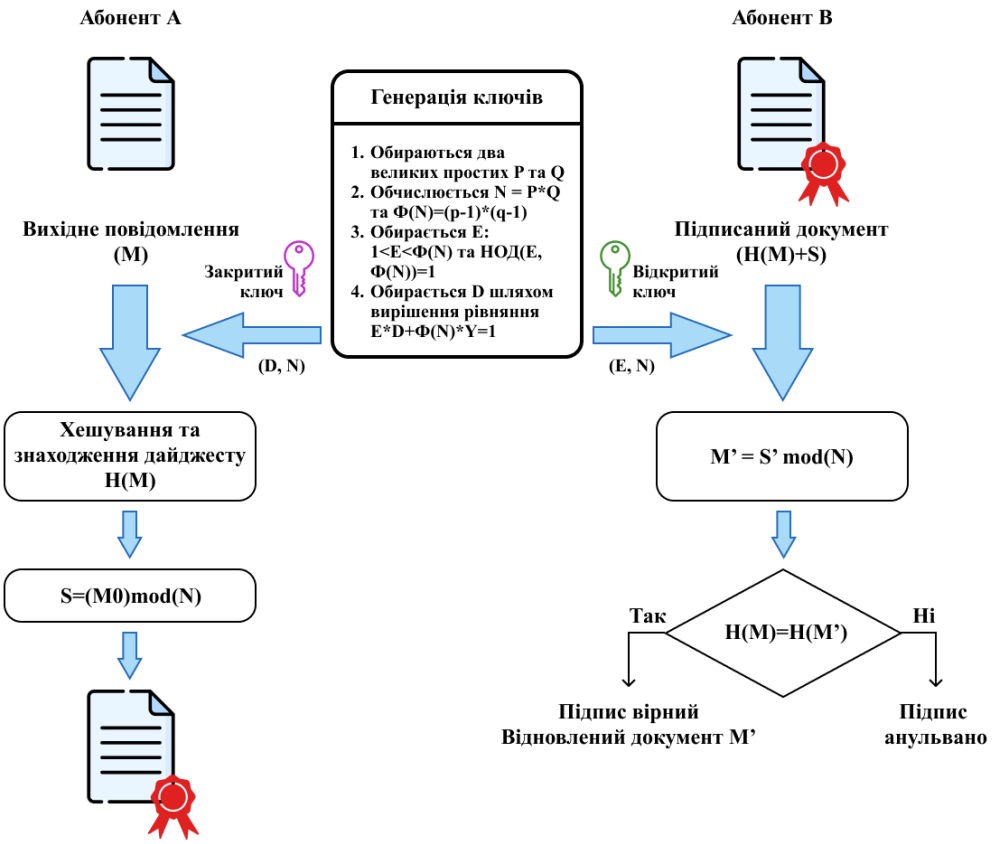


Рис 3. Схема RSA

1.3.2 Схема Ель-Гамаля

Схема Ель-Гамаля — криптосистема з відкритим ключем, яку засновано на складності обчислення дискретних логарифмів у скінченному полі. Криптосистема включає в себе алгоритм шифрування і алгоритм цифрового підпису [10].

Генерація ключової пари.

1) Обирається будь-яке (правда, досить велике) просте число p.

2) Для нього визначається довільний примітивний (утворюючий) елемент – число g.

3) Генерується будь-яке випадкове число x – закритий ключ.

4) Визначається значення $y = g^x \ mod$ p.

5) Комбінація (g, p, y) є відкритим ключем одержувача.

Формування підпису.

1) Генерується випадкове число k, унікальне для кожного підписання документа, взаємно просте з числом $(\text{p} - 1)$, НСД

$$(k, \varphi\ (p)) = 1$$

2) Визначається $r = (g^k \ mod$ p).

3) Визначається $s = ((h - \text{x} * r) / k) \ mod\ (\text{p} - 1)$,

де h - контрольна сума документу, що підписується, пара чисел (r, s) є ЕЦП для документа, що має контрольну суму h.

Перевірка (верифікація) підпису.

Обчислюється $u = ((y^r) * (r^s) \ mod$ p).

Обчислюється $h = hash\ (m)$ і $v = (g^h \ mod$ p)

Перевіряється рівність значень u = v (якщо рівність виконується, то підпис вірний, в іншому випадку документ підроблений).

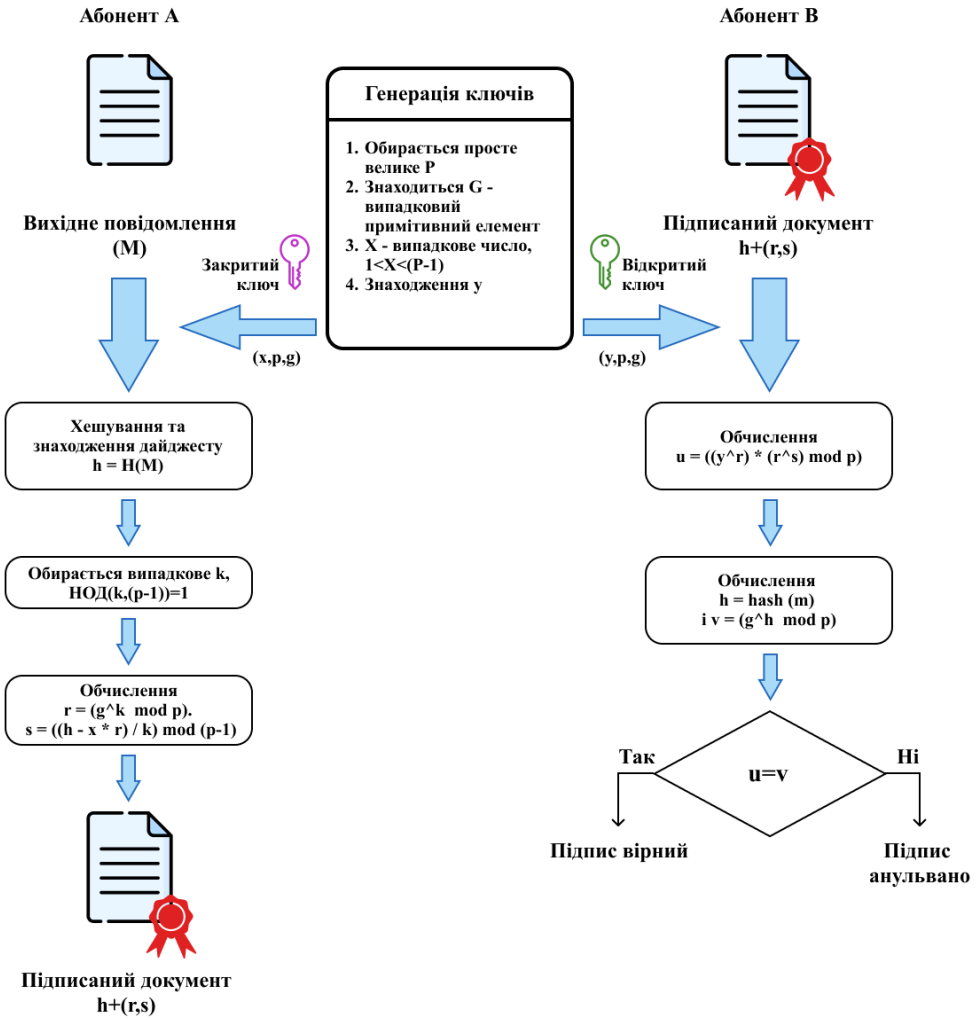


Рис 4. Схема Ель-Гамаля

1.3.3 Digital Signature Algorithm

DSA - криптографічний алгоритм з використанням відкритого ключа для створення електронного підпису, але не для шифрування [11]

Параметри.

Відкритий ключ (p, q, g, y)

p - просте число довжиною від 512 до 1024 бітів;

q - 160-бітовий простий множник $p-1$;

$g = h^{(p-1)/q} \bmod p$,

де h - довільне число, $h < (p-1)$,

для якого $h^{(p-1)/q} \bmod p > 1$;

$y = g^x \bmod p$ (p-бітове число)

Закритий ключ (x)

$1 < x < q$ (160-бітове число)

Формування підпису.

Відправник створює довільне число k, менше q

Відправник створює

$$r = (g^k \bmod p) \bmod q$$

$$s = (k^{-1} (H(m) + x\, r)) \bmod q,$$

де $H(m)$ – хеш-функція

Відправник надсилає підпис - (r, s)

Перевірка (верифікація) підпису.

Одержувач перевіряє підпис, роблячи обчислення

$$w = s^{-1} \bmod q$$

$$u_1 = (H(m) * w) \bmod q$$

$$u_2 = (r\, w) \bmod q$$

$$v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$$

Якщо $v = r$, то підпис вірний

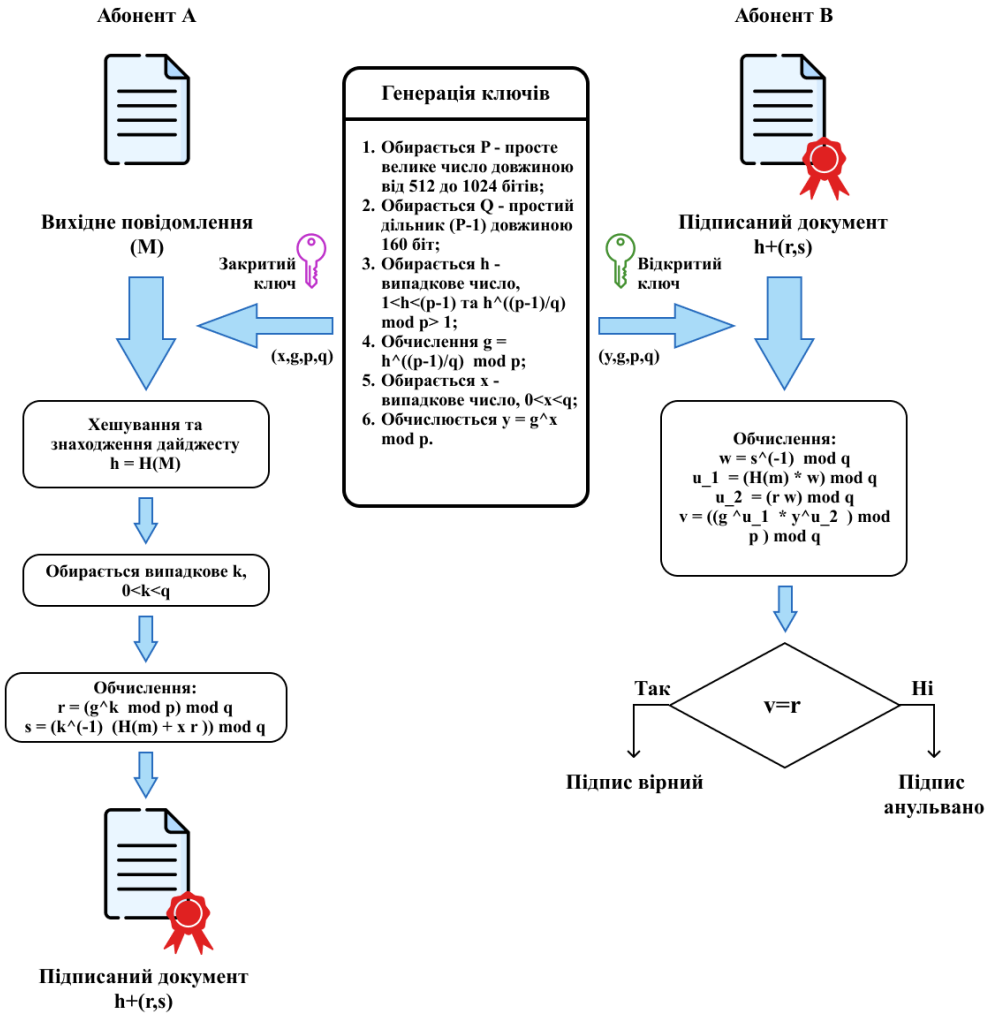


Рис 5. Алгоритм DSA

1.3.4 Elliptic Curve Digital Signature Algorithm

ECDSA - алгоритм з відкритим ключем для створення цифрового підпису, аналогічний за своєю будовою DSA, але визначений, на відміну від нього, не над кільцем цілих чисел, а в групі точок еліптичної кривої [12].

Генерація ключової пари.

1) Вибираємо еліптичну криву E, визначену на Z_p. Кількість точок в $E(Z_p)$ має ділитися на велике ціле n;

2) Обираємо точку $P \in E(Z_з)$ порядку n;

3) Обираємо будь-яке число $d \in [1, n - 1]$;

4) Обчислюємо $Q = dP$;

5) Секретним ключем робимо (d, E, P, n), відкритим – $(\mathrm{E}, \mathrm{P}, n, Q)$.

Формування підпису.

1) Обираємо довільне число $k \in [1, n - 1]$;

2) Обчислюємо $k\mathrm{P} = (\mathrm{x}_1, \mathrm{y}_1)$ та $r = \mathrm{x}_1 \ (mod\ n)$. Якщо $r \neq 0$,

переходимо до кроку 3, в іншому випадку – повертаємося до кроку 1;

3) Обчислюємо $k^{-}l \ (mod\ n)$;

4) Обчислюємо $s = [k^{-l}\ (h(M) + d \times r)] mod\ n$. Якщо $s \neq 0$,

переходимо до кроку 5, в іншому випадку –повертаємося до кроку 1;

5) Підписом під повідомленням M є пара цілих чисел (r, s).

Перевірка (верифікація) підпису.

1) Якщо r і s – цілі числа, належать до інтервалу $[1, n - 1]$,

переходимо до кроку 2, в іншому випадку – результат перевірки є негативний (підпис анулюється);

2) Обчислюємо $w = s^{-l}(mod\ n)$ та $h(M)$;

3) Обчислюємо $u_1 = [h(\mathrm{M}) \times w]\ mod\ n$ та $u_2 = (r \times w)\ mod\ n$;

4) Обчислюємо $u_1 = \mathrm{P} + u_2 Q = (\mathrm{x}_0, \mathrm{y}_0)$ та $v = x_0\ (mod\ n)$;

5) Підпис є справжній за $v = r$.

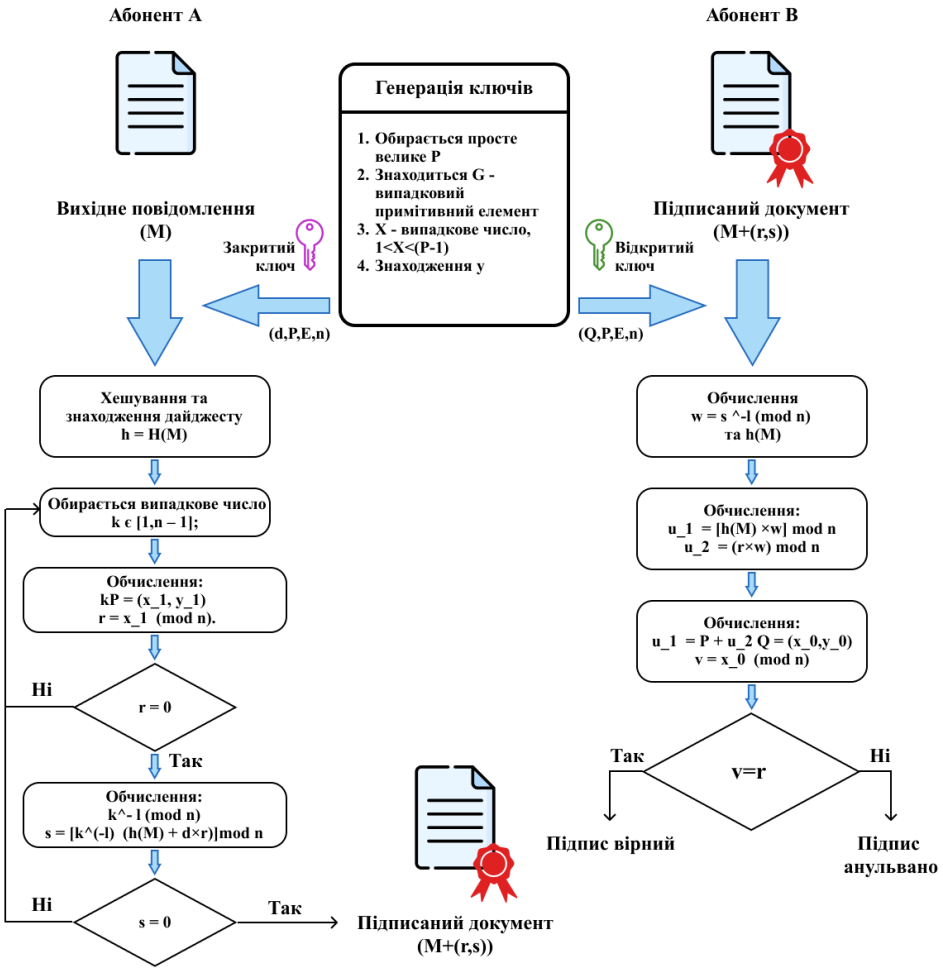


Рис 6. Алгоритм ECDSA

1.3.5 Порівняння криптографічних методів формування та перевірки електронного цифрового підпису

Таблиця 1. Порівняння криптографічних методів

Алгоритм	Проблема, що лежить в основі	Довжина підпису	Розмір відкритого ключа, біт	Хеш-функція або інші функції	Здатність до відновлення повідомлення
RSA	ПФЧ	Довжина ключа *2	1024-4096	MD	Так

Ель-Гамаля	ПДЛ	Довжина ключа *2	1024-4096	MD та SHA	Ні
DSA	ПДЛ	Розмір скінченого поля *4	1024-3072	SHA	Ні
ECDSA	ПДЛЕК	Розмір скінченого поля *4	112-570	SHA	Ні

Таблиця 2. Порівняння швидкості роботи алгоритмів ЕЦП

Алгоритм	Розмір ключа, біт	Швидкість підпису, мс	Швидкість перевірки, мс
RSA	1024 2048	1.48 6.05	0.07 0.16
DSA	1024	0.42	0.52
El-Gamal	1024 2048	0.45 0.83	1.18 3.84
ECDSA над GF(p)	256	2.88	8.53

1.4 Висновки до першого розділу

У першому розділі дипломної роботи були розглянуті вразливості інформації та методи захисту даних. До основних способів належать:

- Методи захисту даних, які обмежують доступ
- Організаційні методи захисту даних
- Технічні методи захисту даних
- Криптографічні методи захисту даних

Також було розкрито питання електронного цифрового підпису, його різновидів, механізму роботи та використання ЕЦП. Серед різновидів були виділені три основні:

- Простий електронний цифровий підпис
- Вдосконалений некваліфікований електронний цифровий підпис
- Вдосконалений кваліфікований електронний цифровий підпис

Станом на сьогодні метод захисту даних на основі ЕЦП є актуальним та ефективним способом захисту інформації.

РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРАНОГО АЛГОРИТМУ ПІДПИСУ

Однією з проблем правових питань в час інформаційних технологій є захист персональних даних. У світі, поглиненому глобалізацією, інформацією про користувача однієї країни користуватися треті особи звідусіль (зазвичай незаконно).

Саме тому питання захисту інформації є невід'ємною частиною сучасного суспільства. Опираючись на висновки, зроблені в першому розділі дипломної роботи, бачимо, що одним з найкращих методів захисту даних є метод захисту з використанням ЕЦП.

Після проведеного аналізу алгоритмів, було визначено, що для захисту даних найбільше підходить алгоритм DSA. Саме цей алгоритм буде програмно реалізовуватись далі у роботі.

2.1 Опис засобів для створення програмного модуля

Для програмної реалізації обраного алгоритму підпису потрібно використати інтегроване середовище розробки для мов програмування C/C++ та редактор вихідних кодів Visual Studio Code.

2.1.1 Інтегроване середовище розробки DEV C++

Dev-C ++ - це повнофункціональне інтегроване середовище розробки C і C ++ (IDE) для платформ Windows. Мільйони розробників, студентів та дослідників використовують Dev-C ++ з моменту випуску першої версії в 1998 році. Він був представлений у десятках C ++ та наукових книгах і залишається одним з улюблених засобів навчання серед університетів та шкіл у всьому світі.

Список особливостей:

- Надзвичайно легкий і портативний IDE C / C ++ для систем Windows
- Підтримує компілятори на основі GCC (Mingw, Cygwin, ...)
- Швидко створюйте графічний інтерфейс та консольні програми Windows, статичні бібліотеки та бібліотеки DLL
- Інтегрований налагоджувач
- Браузер класів
- Завершення коду
- Перелік функцій
- Підтримка профілювання
- Доступно понад 30 мов
- Настроюваний редактор коду
- Керівник проекту
- Шаблони для створення власних типів проектів
- Генерація файлу Makefile
- Редагуйте та компілюйте файли ресурсів
- Менеджер інструментів
- Знайти та замінити об’єкти
- Підтримка CVS [13]

2.1.2 Visual Studio Code

Visual Studio Code - це легкий, але один з найкращих редакторів вихідних кодів, він доступний для Windows, macOS та Linux.

Він працює із вбудованою підтримкою JavaScript, TypeScript та Node.js і включає велику кількість розширень для інших мов (таких як C ++, C #, Java, Python, PHP, Go) та середовищ виконання (наприклад .NET та Unity) .

До переваг Visual Studio Code можна віднести :

- Хороші вбудовані функції, такі як автоматичне підсвічування повторюваних змінних
- Легкість використання
- Швидка модифікація сценарію

У своїй основі Visual Studio Code має відмінний редактор вихідного коду, ідеально підходить для повсякденного використання. Завдяки підтримці багатьох мов, VS Code допомагає підвищити продуктивність за допомогою підсвічування синтаксису, відповідності дужок, автоматичного відступу, виділення вікон, фрагментів тощо. Інтуїтивно зрозумілі комбінації клавіш, просте налаштування та відображення комбінацій клавіш, внесених спільнотою, дозволяють легко орієнтуватися в коді.

Для серйозного кодування ви часто отримуєте користь від інструментів, які більше розуміють код, ніж просто текстові блоки. Visual Studio Code включає вбудовану підтримку заповнення коду IntelliSense, розширене розуміння семантичного коду та навігацію, а також рефакторинг коду.

VS Code також інтегрується з інструментами побудови та сценарію для виконання загальних завдань, що пришвидшує щоденні робочі процеси. VS Code має підтримку Git, тому ви можете працювати з керуванням джерела, не виходячи з редактора, включаючи перегляд змін [14].

2.1.3 Мова програмування C++

C ++ - це міжплатформна мова, яку можна використовувати для створення високопродуктивних додатків. C ++ була розроблений Bjarne Stroustrup як розширення мови C. C ++ надає програмістам високий рівень контролю над системними ресурсами та пам'яттю. Мова була оновлена 3 основних рази в 2011, 2014 та 2017 роках до C ++ 11, C ++ 14 та C ++ 17 [15].

Бібліотеки, які будуть використані в програмі:

<iostream> - бібліотека введення / виводу. Виводить кілька стандартних об'єктів потоку

<stdlib.h> - поводиться так, ніби кожне ім'я з <cstdlib> розміщено у глобальному просторі імен

<sstream> - бібліотека введення / виводу для std::basic_stringstream, std::basic_istringstream, std::basic_ostringstream класів

<fstream> - бібліотека введення / виводу для std::basic_fstream, std::basic_ifstream, std::basic_ofstream класів

<string> - бібліотека рядків. Шаблон класу std :: basic_string

<math.h> - поводиться так, ніби кожне ім'я з <cmath> розміщено у глобальному просторі імен, крім назв математичних спеціальних функцій

<ctime> - утиліти часу / дати

<locale> - бібліотека локалізації [16]

```
1   #include <iostream>
2   #include <stdlib.h>
3   #include <sstream>
4   #include <fstream>
5   #include <conio.h>
6   #include <string>
7   #include <math.h>
8   #include <ctime>
9   #include<locale>
10  #include <Windows.h>
```

Рис 7. Використані для реалізації алгоритму DSA бібліотеки

2.1.4 JavaScript

JavaScript (часто скорочується до JS) - це легка, інтерпретована об'єктно-орієнтована мова з першокласними властивостями і найбільш відома як мова сценаріїв веб-сторінок, але також використовується у багатьох середовищах, що не належать до браузера. Це мова сценаріїв на основі прототипу, яка

використовує ряд парадигм, які є динамічними та підтримують об’єктно-орієнтовані, необхідні та функціональні стилі програмування.

JavaScript працює на веб-сайті клієнта, який може бути використаний для проектування / програмування поведінки веб-сайту у випадку події. JavaScript - це легка для вивчення та потужна мова сценаріїв, яка часто використовується для контролю поведінки веб-сайтів [17].

JavaScript може виступати як процедурна та об'єктно-орієнтована мова. Об'єкти створюються програмно в JavaScript, а методи та властивості додаються до порожніх об'єктів під час виконання, на відміну від визначень класів синтаксису, поширених у перекладених мовах, таких як C ++ та Java. Як тільки об’єкт побудований, його можна використовувати як план (або прототип) для створення подібних об’єктів.

Динамічні можливості JavaScript включають створення виконуваного об’єкта, списків змінних, функціональних змінних, створення динамічного сценарію (через eval), інтроспективного об’єкта (через for ... in) та відновлення вихідного коду (програми JavaScript можуть розкладати тіла функцій назад у вихідний текст).

2.2 Програмна реалізація алгоритму методу формування цифрового підпису

Відповідно до алгоритму були задані початкові значення p, q, x, k, H, де

p – просте число, $2^{L-1} < p < 2^L$, де $512 < L < 1024$ і L кратне 64

q – простий дільник $p - 1$, довжиною 160 бітів;

x – випадкове або псевдовипадкове число, $0 < x < q$.

k - випадкове або псевдовипадкове число, $0 < k < q$.

```
#include <iostream>
#include <stdlib.h>
#include <sstream>
#include <fstream>
#include <conio.h>
#include <string>
#include <math.h>
#include <ctime>
#include<locale>
#include <Windows.h>
using namespace std;

int p = 569, q = 71, x = 6, k = 2, H = 668;
```

Рис 8. Оголошення зміних

Оголошення функцій для перевірки h на відповідність умові: h - ціле, $1 < h < p - 1$ і $h^{(p-1)/q} \ mod \ p > 1$, та для вводу h користувачем

```
bool check_h(int h) {
   return h > 1 && h < p - 1 && fmod(pow(h, (p - 1) / q), p) > 1;
}

int get_h() {
    int h;
    do {
        cout << endl << "Введіть число h: ";
        cin >> h;
    } while (!check_h(h));
    return h;
}
```

Рис 9. Оголошення функцій

Основна робота програми, де відповідно до алгоритму обчислюється g – глобальна компонента відкритого ключа, y – відкритий ключ користувача. Далі, за допомогою оголошених на початку зміних та обчислених g,y, створюється підпис. Визначаються зміні r, s, відповідно до формул DSA алгоритму, та він виводиться на екран.

```
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(LC_CTYPE,"ukr");
    clock_t start = clock();
//      int h = 20;
    int h = get_h();
//  while (true) {

    cout << "h = " << h << endl;

    int g = fmod(pow(h, (p - 1) / q), p);
    cout << "g = " << g << endl;
    cout << "x = " << x << endl;

    int y = fmod(pow(g, x), p);
    cout << "y = " << y << endl;
    cout << "k = " << k << endl;

    cout << endl << "Створення підпису: " << endl;

    int r = fmod(y, q);
    cout << "r = " << r << endl;

    cout << "Хеш H(M) = " << H << endl;
    int s = fmod(pow(k, -1) * (H + r * x), q);
    cout << "s = " << s << endl;

    cout << "Підпис: (" << r << ", " << s << ")" << endl;
```

Рис 10. Створення підпису

Проводиться перевірка (верифікація) підпису за допомогою обчислення за формулами алгоритму DSA зміних – w, u_1, u_2, v. Перевіряється умова $v = r$, з якої робиться висновок верифіковано чи не верифіковано. Визначається час роботи програми.

```
    cout << endl << "Верифікація підпису: " << endl;

    int w = fmod(s, q);
    int u1 = fmod(H * w, q);
    int u2 = fmod(r * w,q);
    int v = fmod(fmod(pow(g, u1) * pow(y, u2), p), q);
    cout << "v = " << v << ", r = " << r << endl << endl;

    string msg =  v == r ? "Верифіковано!" : "Не верифіковано!";
    cout << msg;
//    h++;
//    if (v == r) break;
//}

    clock_t end = clock();
    double seconds = (double)(end - start) / CLOCKS_PER_SEC;

    cout << endl << endl << "Час проведення процедури формування/верифікації ЕЦП: " << seconds<< " сек." << endl << endl;
    return 0;
}
```

Рис 11. Верифікація підпису

Робота програмно реалізованого алгоритму DSA зі створення та верифікації ЕЦП. У даному випадку $v \neq r$ (Рис.12), отже підпис анульовано.

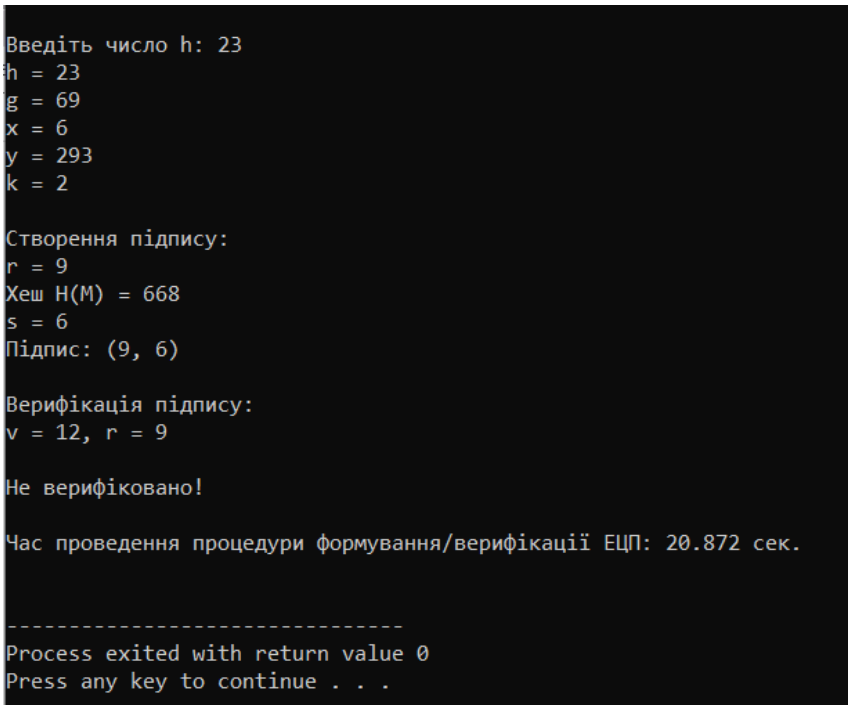


Рис 12. Робота програми. Варіант без верифікації.

Робота програмно реалізованого алгоритму DSA зі створення та верифікації ЕЦП. У даному випадку $v = r$ (Рис.12), отже підпис вірний.

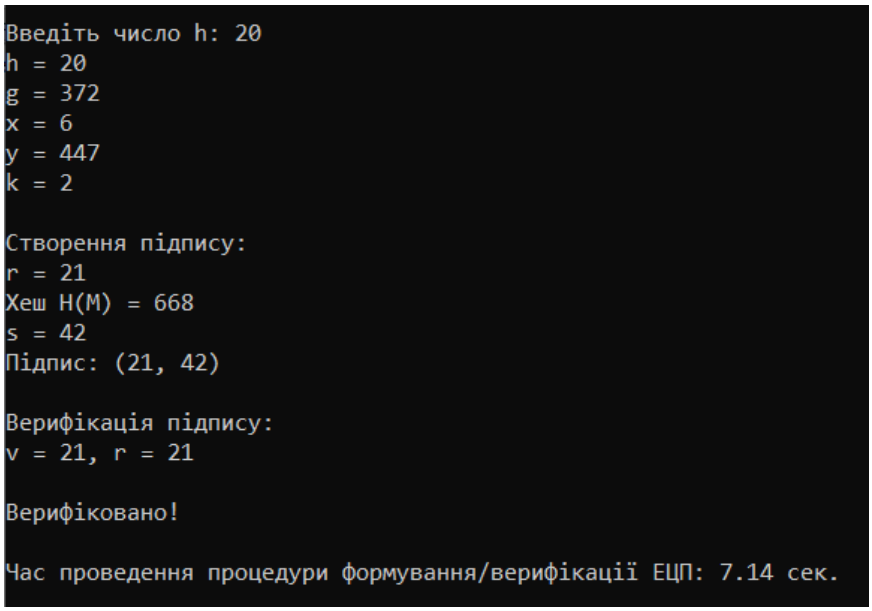


Рис 13. Робота програми. Варіант з успішною верифікацією.

2.3 Програмний модуль створення методу захисту даних з використанням електронного цифрового підпису

За допомогою редактору вихідних кодів VS Code програмно реалізується генерація ключів, використовуючи алгоритм DSA. Для створення відкритого ключа в коді використовується тип – spki. Це спеціальна інфраструктура відкритих ключів для подолання недоліків стандарту X.509 [18].

Для створення особистого ключа використовується тип – pkcs8, який в сучасній криптографії є стандартом синтаксису для зберігання приватних ключів [19].

Рис 14. Програмний код генерації ключів

Далі для реалізації ЕЦП потрібно створити функцію підпису документу.

```
JS signer.js > ...
1   const crypto = require('crypto');
2   const fs = require('fs');
3
4   module.exports.sign = filename => {
5     const private_key = fs.readFileSync('keys/privateKey.pem', 'utf-8');
6
7     const doc = fs.readFileSync('uploads/' + filename);
8
9     const signer = crypto.createSign('DSA-SHA256');
10    signer.write(doc);
11    signer.end();
12
13    const signature = signer.sign(private_key, 'hex');
14
15    console.log('Digital Signature: ', signature);
16
17    fs.writeFileSync('signature.hex', signature);
18    return signature;
19  };
20
```

Рис 15. Програмний код підпису документу

Останнім кроком реалізації обраного алгоритму є верифікація створеного раніше в програмі підпису.

```
JS verify.js > ...
1   const crypto = require('crypto');
2   const fs = require('fs');
3
4   module.exports.verify = (filename, isKeyUploaded, isSignatureUploaded) => {
5     const publicKeyFolder = isKeyUploaded ? 'uploads/' : 'keys/';
6     const signatureFolder = isSignatureUploaded ? 'uploads/' : '';
7
8     let public_key = fs.readFileSync(`${publicKeyFolder}publicKey.pem`, 'utf-8');
9
10    const signature = fs.readFileSync(`${signatureFolder}signature.hex`, 'utf-8');
11
12    const doc = fs.readFileSync('uploads/' + filename);
13
14    const verifier = crypto.createVerify('DSA-SHA256');
15    verifier.write(doc);
16    verifier.end();
17
18    const result = verifier.verify(public_key, signature, 'hex');
19
20    console.log('Digital Signature Verification : ' + result);
21
22    return result;
23  };
24
```

Рис 16. Програмний код верифікації документу

Для перевірки ефективності створеного методу захисту даних потрібно провести тестування створеного додатку за наступним алгоритмом:

– Завантаження документу для підпису
– Генерація ключової пари
– Підпис завантаженого документа
– Верифікація підпису
– Внесення змін в документ (атака)
– Повторна верифікація

Для початку перевірки відкриваємо початкове меню створеного додатку

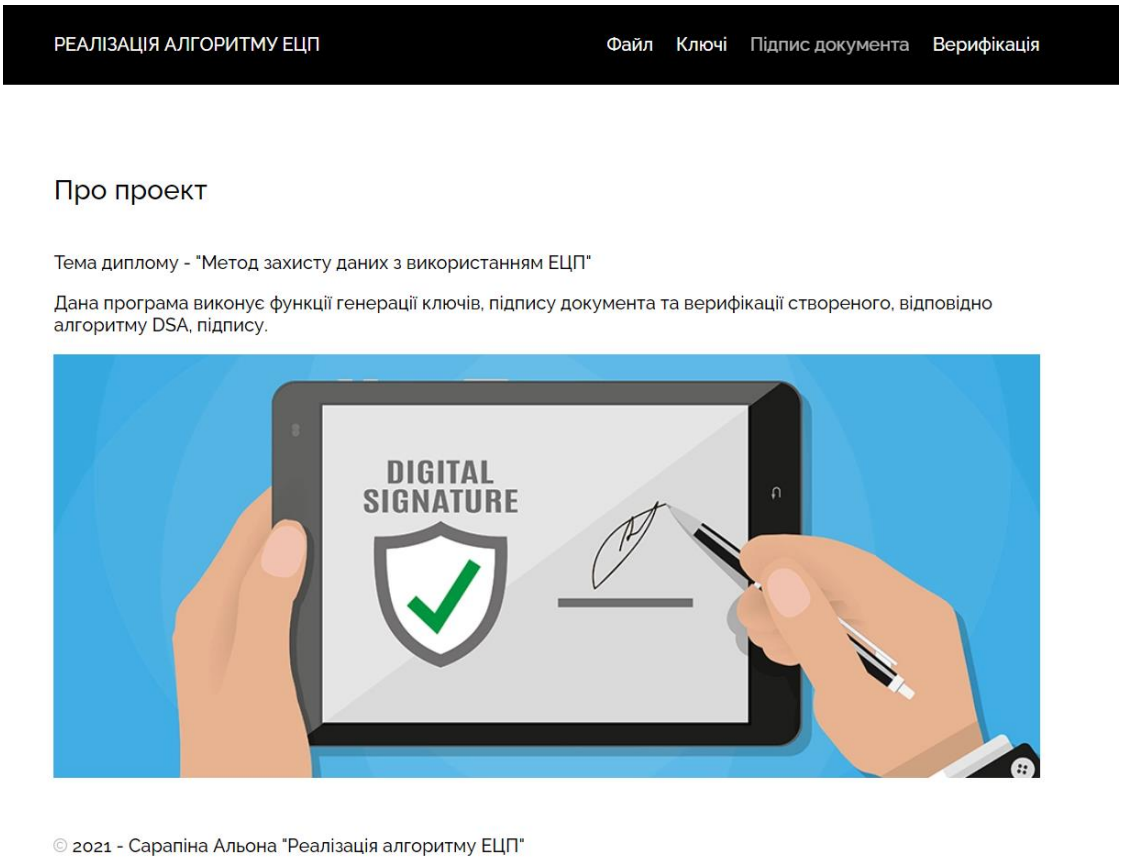


Рис 17. Початкове меню додатку

Далі виконуємо перший пункт алгоритму, а саме завантаження документу для його захисту за допомогою накладання ЕЦП. За допомогою меню «Файл» виконується імпорт текстового файлу.

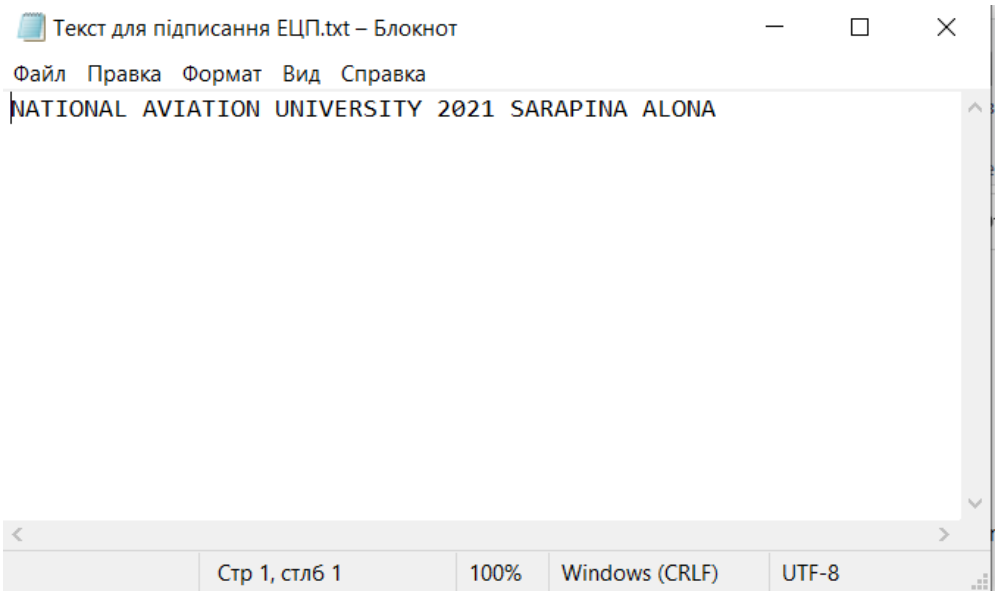


Рис 18. Текст для накладання ЕЦП

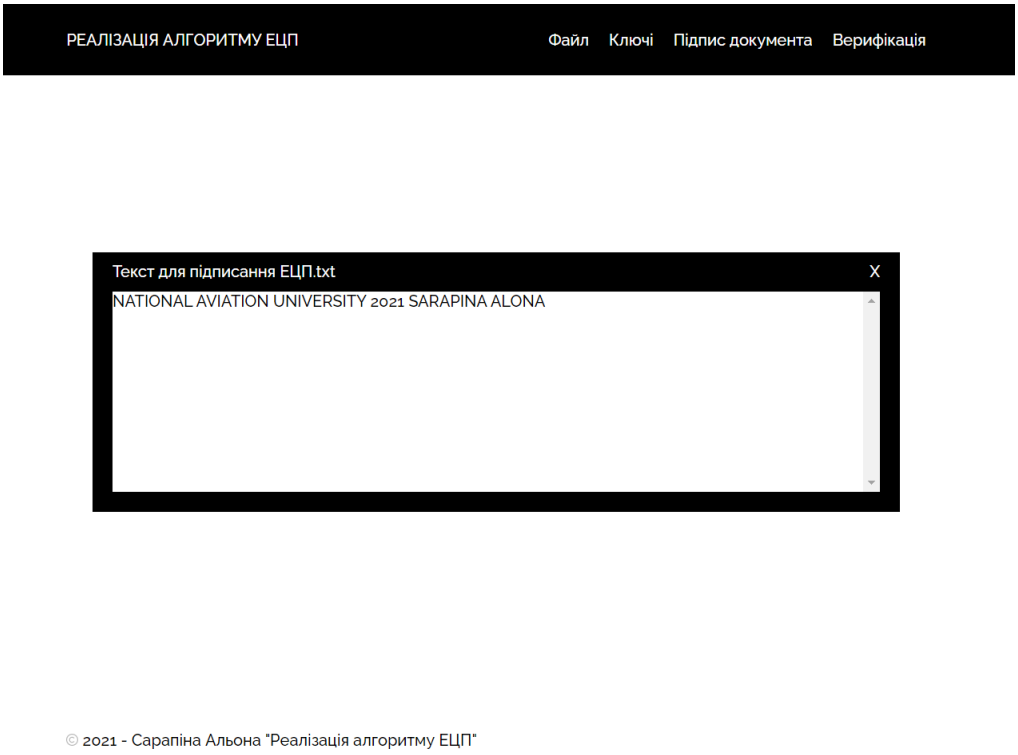


Рис 19. Завантаження тексту

Після завантаження тексту, потрібно згенерувати особистий та публічний ключ для їх подальшого використання.

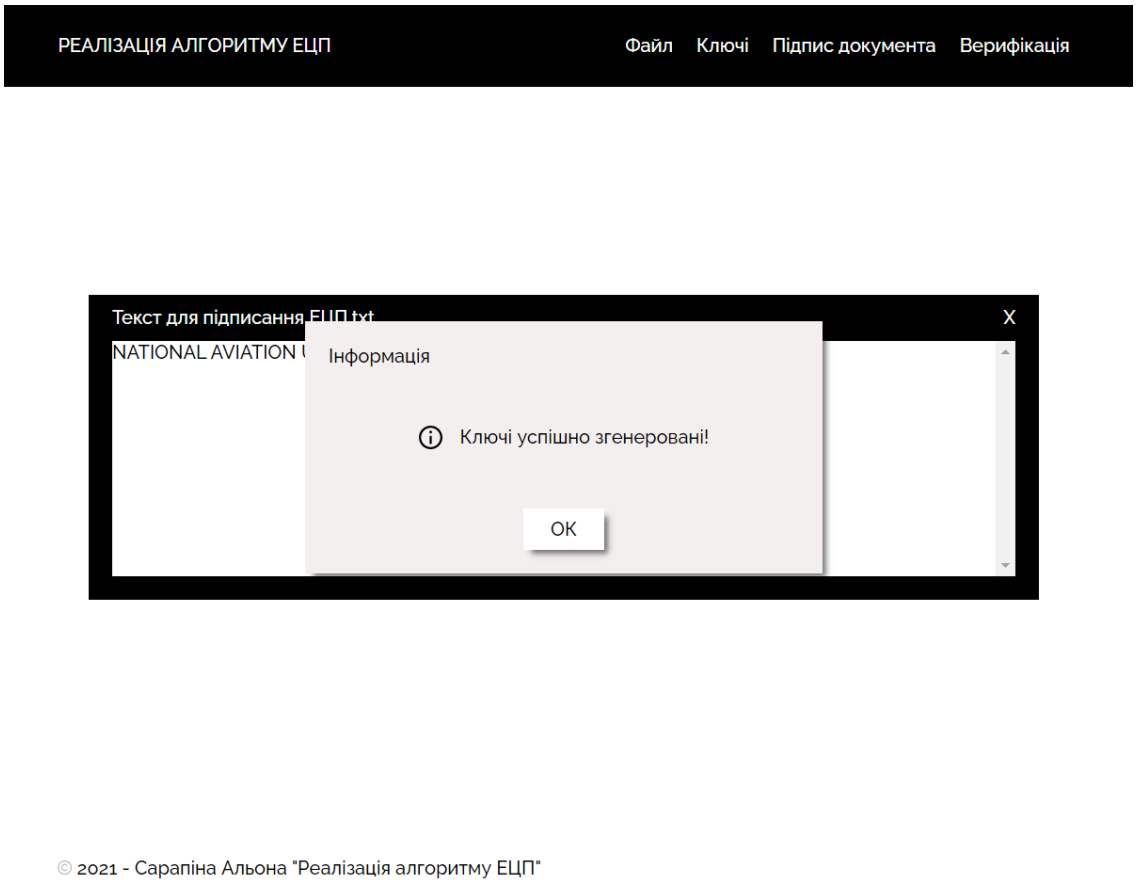


Рис 20. Генерація ключів

Після генерації, ключі зберігаються для можливості їх перегляду та передачі.

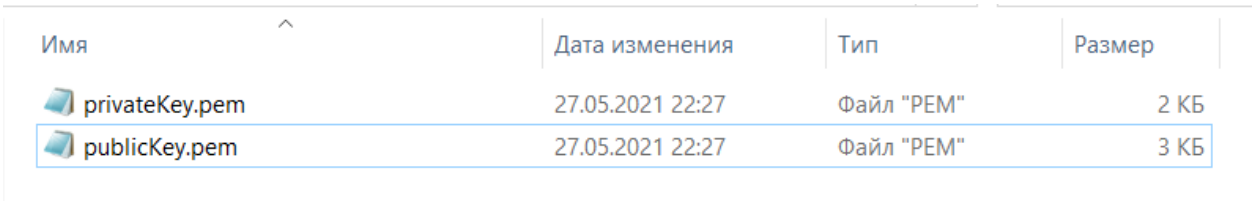


Рис 21. Збереженні ключі

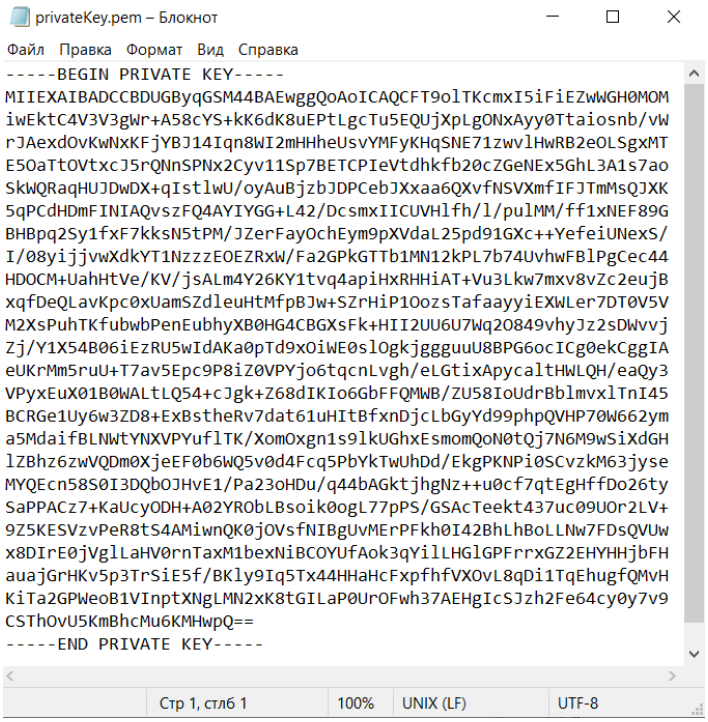


Рис 22. Приватний ключ

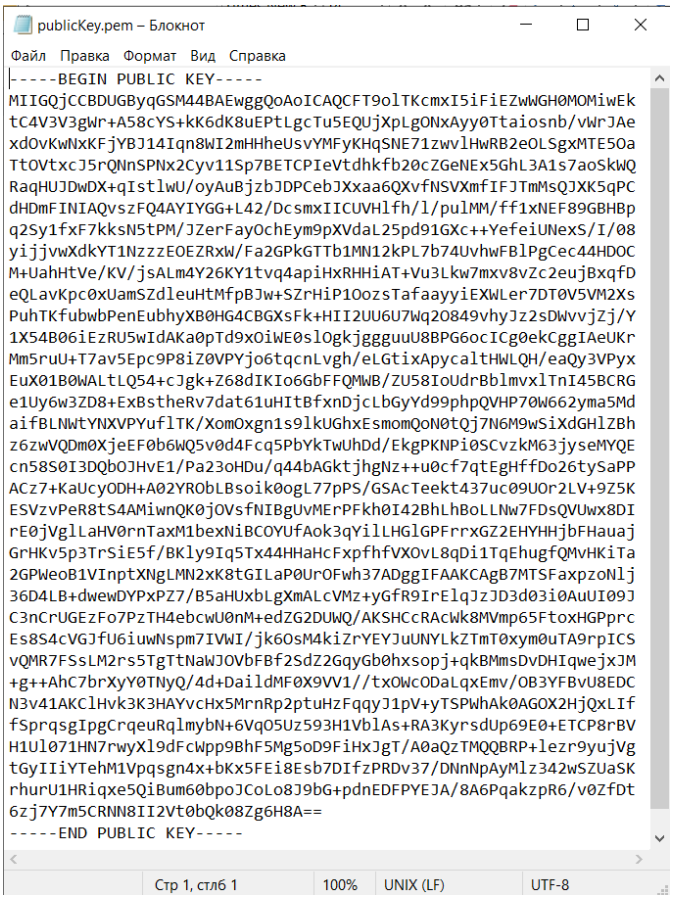


Рис 23. Публічний ключ

Після генерації ключової пари виконується підписання завантаженого документа.

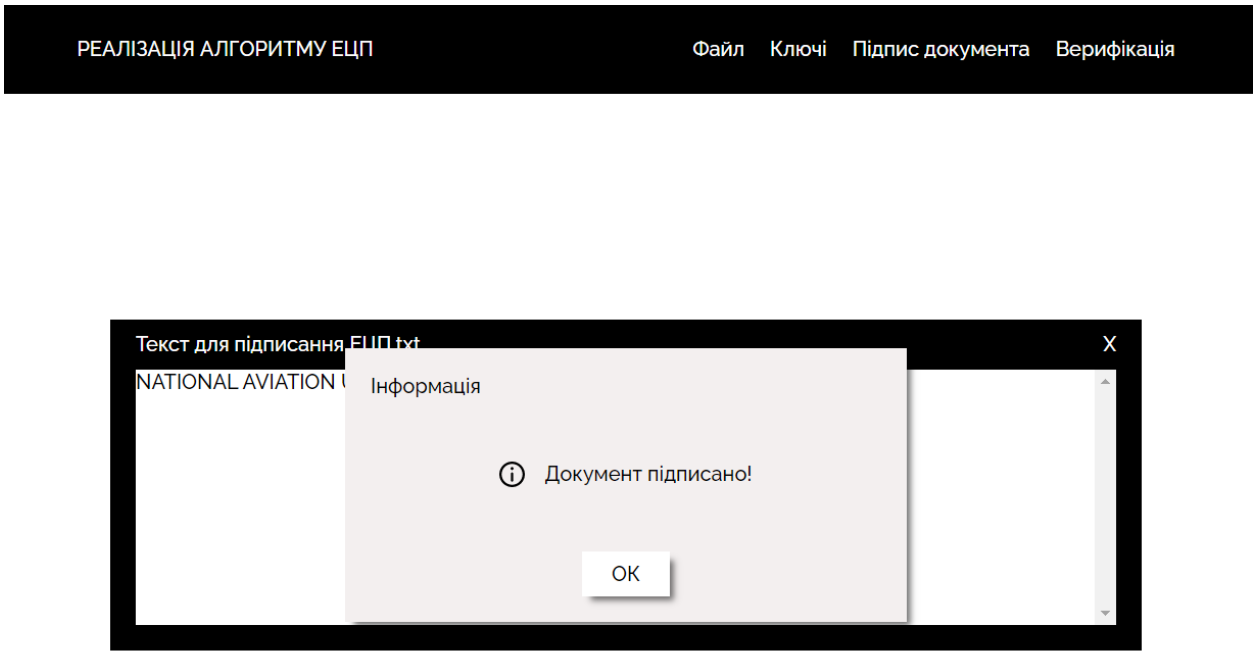


Рис 24. Підписання документу

Після підписання документу, підпис зберігається в головну папку для подальшої верифікації

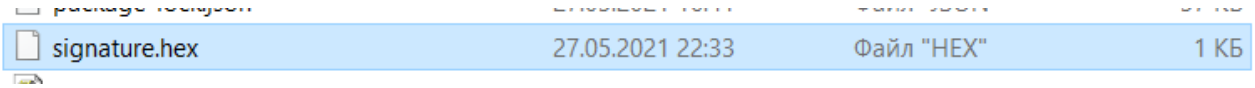


Рис 25. Створений електронний цифровий підпис

Проводиться верифікація підпису для перевірки внесення змін у документ.

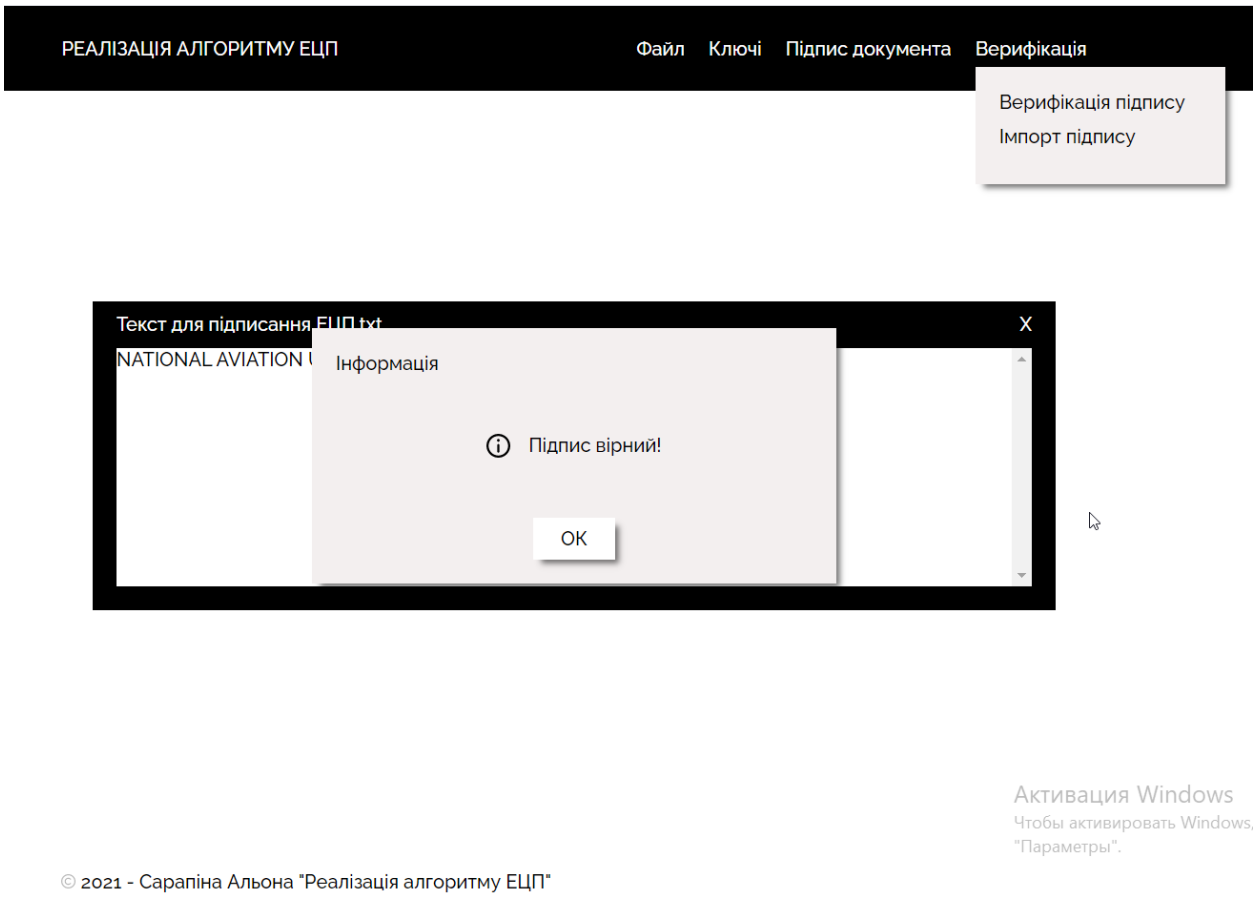


Рис 26. Верифікація створеного підпису

Далі редагується вміст завіреного текстового файлу для перевірки надійності електронного цифрового підпису.

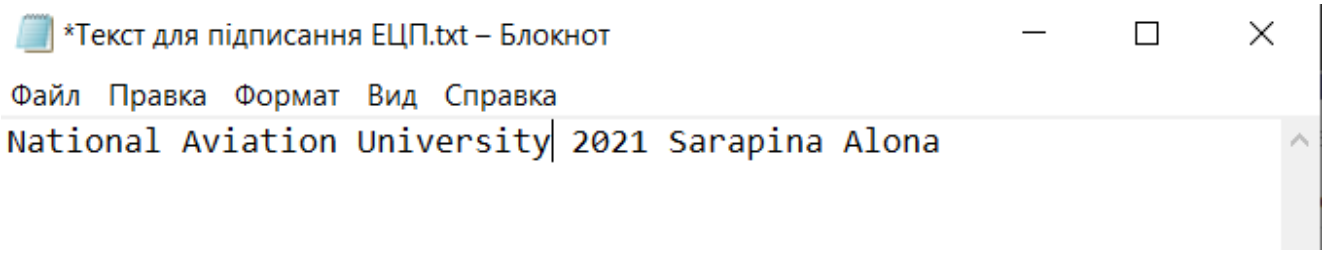


Рис 27. Редагування тексту

Останнім кроком перевірки є повторна верифікація зміненого документу, для виявлення змін.

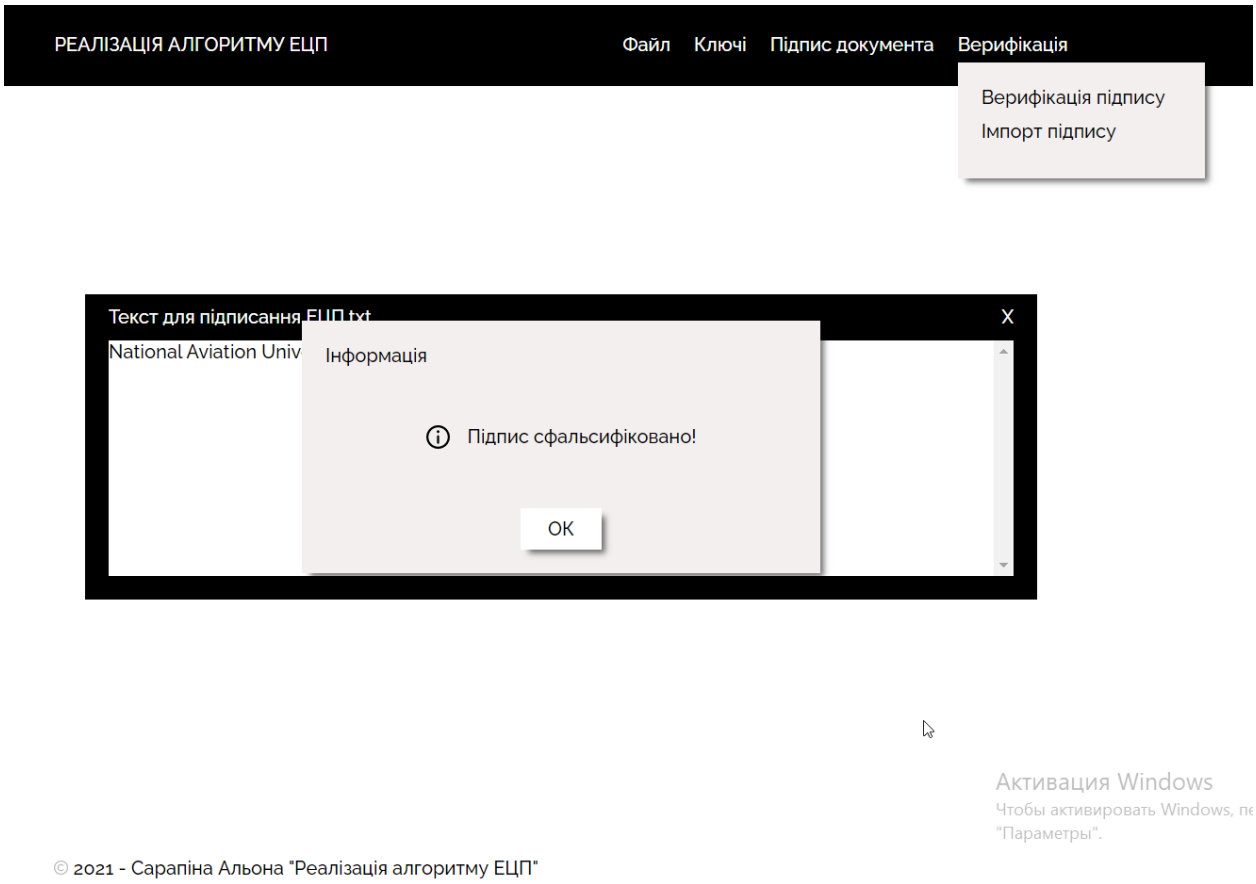


Рис 28. Результат атаки

2.4 Висновки до другого розділу

У другому розділі дипломної роботи було реалізовано програмно обраний у першому розділі алгоритм підпису. Також було проведено генерацію ключів , підписання документа обраною схемою, верифікацію ЕЦП до та після внесення змін в документ.

Для програмного створення DSA алгоритму , було використано інтегроване середовище програмування для мов розробки C/C++ - Dev C++.

Переваги даного застосунку в легкості його використання. Дане середовище програмування має простий та приємний для користувача інтерфейс.

Також для створення ЕЦП, за допомогою реалізованого вище алгоритму, інтерсейсу і функціональності веб-додатку було використано редактор вихідник кодів – VS Code, який підтримує сотню мов, зокрема JavaScript.

Код реалізації алгоритму створено за допомогою мови C++. Дану мову програмування використовують для створення проектів, в яких є пріотрітетною продуктивність коду. До переваг даної мови можна віднести динамічну обробку інформації та високу продуктивність компіляторів.

Код для коректної роботи ЕЦП, інтерфейс та функціональність веб-додатку створено за допомогою мови JavaScript. До переваг даної мови можна віднести : швидкість роботи, простоту використання, широкий інтерфейс, розширену функціональність та універсальність.

Шляхом створення програмного модуля обраного алгоритму підпису, у другому розділі дипломної роботи, було доведено, що електронний цифровий підпис - є надійним способом захисту інформації. Завдяки ньому зловмисник не зможе вносити зміни в створений офіційний документ , а отже його цілісність не постраждає. Отже , створений метод захисту даних з використанням ЕЦП можна вважати актуальним.

ВИСНОВКИ

Інформаційна галузь постійно змінюється, а отже змінюються і загрози пов’язані з нею. Таке збільшення небезпек вимагає створення нових методів захисту даних.

У дипломній роботі було вирішено одне з найактуальніших питань сучасної інформаційної безпеки, а саме – забезпечення цілісності даних. Було створено метод захисту даних, який базується на використанні електронного цифрового підпису. Під час виконання роботи:

1. Було проведено аналіз існуючих методів захисту даних, який показав, що криптографічні методи захисту є ефективними для забезпечення безпеки інформації.
2. Було здійснено аналіз основних алгоритмів реалізації ЕЦП, а саме : алгоритму RSA, алгоритму Ель-Гамаля, DSA та ECDSA алгоритми, що дало можливість вибору алгоритму електронного цифрового підпису.
3. Було розроблено програмну реалізацію криптографічного алгоритму DSA з використанням інтегрованого середовища розробки для мов програмування C/C++ та редактора вихідних кодів Visual Studio Code, що дало змогу забезпечити цілісність інформаційних ресурсів.

За результатами перевірки додатку, можна зробити висновок, що створений метод захисту даних є надійним, адже при внесенні змін у підписаний документ, програма повідомляє про фальсифікацію підпису.

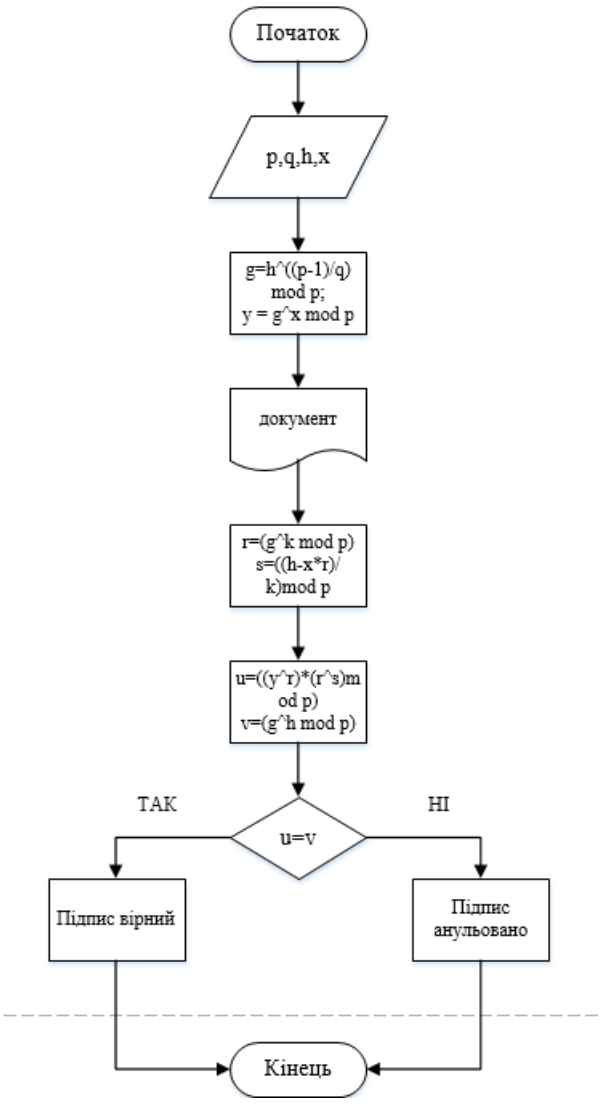
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про захист персональних даних [Текст]: Закон України № 2297-VI від 01 червня 2010 р. / Верховна Рада України // Відомості Верховної Ради України. – 2010. – №34. – Ст. 481.
2. Методы защиты информации [Електронний ресурс] – Режим доступу до ресурсу: https://spravochnick.ru/informacionnaya_bezopasnost/zaschita_informacii/metody_zaschity_informacii/
3. Ісмайлов К.Ю. Криптографічні методи захисту інформації види та вимоги до них // Одеський державний університет внутрішніх справ. – 2018. – С.181
4. Назначение криптографических методов защиты информации [Електронний ресурс] – Режим доступу до ресурсу: http://aguryanov.blogspot.com/2012/09/blog-post_3412.html
5. Електронний цифровий підпис [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Електронний_цифровий_підпис
6. Про електронний цифровий підпис [Текст]: Закон України № 852-IV від 05 жовтня 2003 р. / Верховна Рада України // Відомості Верховної Ради України. – 2003. – №36. – Ст. 276
7. ЕЦП [Електронний ресурс] – Режим доступу до ресурсу: https://www.audit-it.ru/terms/agreements/elektronno_tsifrovaya_podpis_etsp.html
8. Про затвердження форм заяв у сфері державної реєстрації юридичних осіб, фізичних осіб - підприємців та громадських формувань [Текст]: Наказ № 3268/5 від 18 листопада 2016 р.
9. RSA [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/RSA

10. Схема Ель-Гамаля [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Схема_Ель-Гамаля
11. DSA [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/DSA
12. Elliptic Curve Digital Signature Algorithm [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm
13. Home – Dev C++ [Електронний ресурс] – Режим доступу до ресурсу: https://www.bloodshed.net/
14. Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: https://code.visualstudio.com/docs/editor/whyvscode
15. Веб-сайт [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/cpp/cpp_intro.asp
16. Веб-сайт [Електронний ресурс] – Режим доступу до ресурсу: https://en.cppreference.com/w/cpp/header
17. Веб-сайт [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
18. SPKI [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/SPKI
19. PKCS8 [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/PKCS_8

Додаток А

Блок-схема DSA алгоритму



Додаток Б

Програмна реалізація DSA алгоритму

```
#include <iostream>
#include <stdlib.h>
#include <sstream>
#include <fstream>
#include <conio.h>
#include <string>
#include <math.h>
#include <ctime>
#include<locale>
#include <Windows.h>
using namespace std;
int p = 569, q = 71, x = 6, k = 2, H = 668;
bool check_h(int h) {
  return h > 1 && h < p - 1 && fmod(pow(h, (p - 1) / q), p) > 1;
}
int get_h() {
   int h;
   do {
       cout << endl << "Введіть число h: ";
       cin >> h;
   } while (!check_h(h));
   return h;
}
int main() {
          SetConsoleCP(1251);
          SetConsoleOutputCP(1251);
          setlocale(LC_CTYPE,"ukr");
          clock_t start = clock();
```

Продовження додатку Б

```
    int h = get_h();
//        while (true) {

    cout << "h = " << h << endl;

    int g = fmod(pow(h, (p - 1) / q), p);
    cout << "g = " << g << endl;
    cout << "x = " << x << endl;

    int y = fmod(pow(g, x), p);
    cout << "y = " << y << endl;
    cout << "k = " << k << endl;

    cout << endl << "Створення підпису: " << endl;

    int r = fmod(y, q);
    cout << "r = " << r << endl;

    cout << "Хеш H(M) = " << H << endl;
    int s = fmod(pow(k, -1) * (H + r * x), q);
    cout << "s = " << s << endl;

    cout << "Підпис: (" << r << ", " << s << ")" << endl;

    cout << endl << "Верифікація підпису: " << endl;

    int w = fmod(s, q);
    int u1 = fmod(H * w, q);
    int u2 = fmod(r * w,q);
```

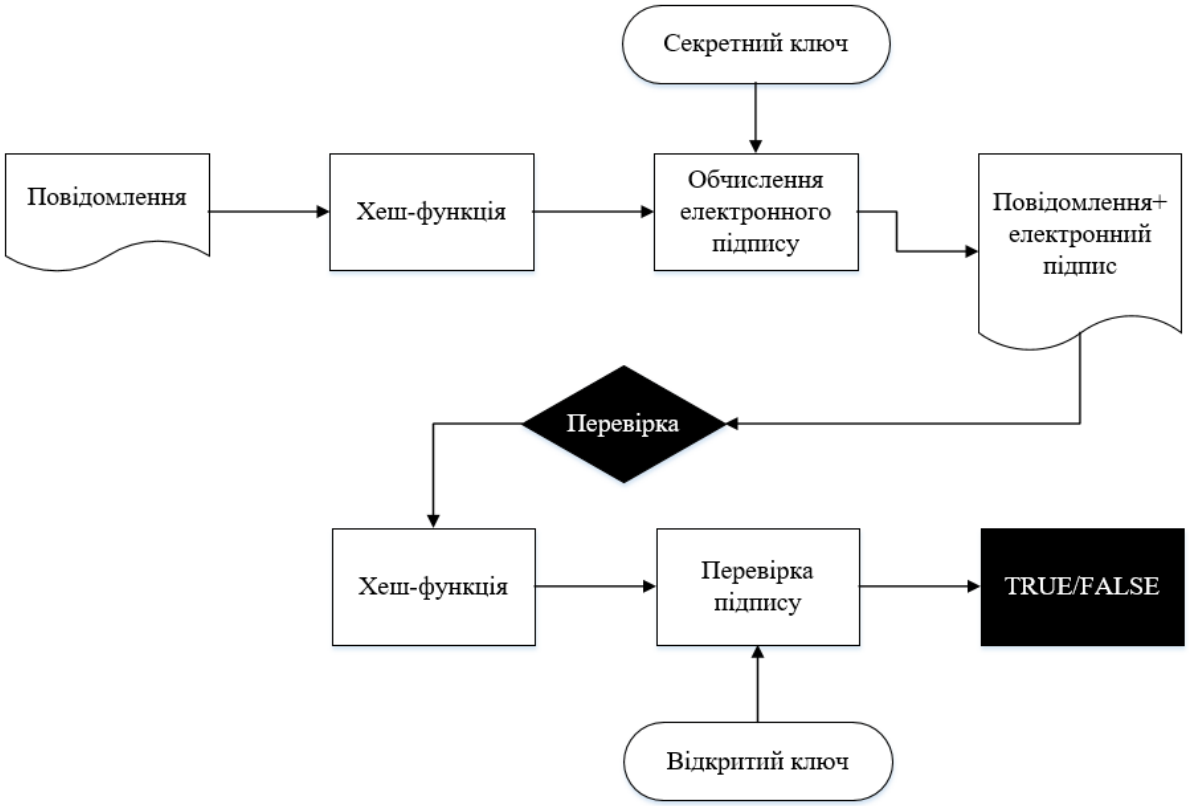
Продовження додатку Б

```
    int v = fmod(fmod(pow(g, u1) * pow(y, u2), p), q);
    cout << "v = " << v << ", r = " << r << endl << endl;
    string msg =  v == r ? "Верифіковано!" : "Не верифіковано!";
    cout << msg;
    clock_t end = clock();
        double seconds = (double)(end - start) / CLOCKS_PER_SEC;


        cout << endl << endl << "Час проведення процедури формування/верифікації ЕЦП: "
<< seconds<< " сек." << endl << endl;
    return 0;
}
```

Додаток В

Алгоритм роботи програмного модулю ЕЦП



Додаток Г

Код програмного модулю генерації ключів

```
const crypto = require('crypto');
const fs = require('fs');
const KEYS_FILE_PATH = 'keys/';
const KEYS_FILE_EXTENSION = '.pem';
module.exports.generateKeys = () => {
  crypto.generateKeyPair(
    'dsa',
    {
      modulusLength: 4096,
      publicKeyEncoding: {
        type: 'spki',
        format: 'pem',
      },
      privateKeyEncoding: {
        type: 'pkcs8',
        format: 'pem',
      },
    },
    (err, publicKey, privateKey) => {
      fs.writeFileSync(`keys/publicKey${KEYS_FILE_EXTENSION}`, publicKey, {
        encoding: 'utf8',
        flag: 'w',
      });
```

Продовження додатку Г

```
    fs.writeFileSync(`keys/privateKey${KEYS_FILE_EXTENSION}`, privateKey, {
      encoding: 'utf8',
      flag: 'w',
    });
  }
);};
```

Додаток Д

Код створення електронного цифрового підпису

```
const crypto = require('crypto');

const fs = require('fs');

module.exports.sign = filename => {

  const private_key = fs.readFileSync('keys/privateKey.pem', 'utf-8');

  const doc = fs.readFileSync('uploads/' + filename);

  const signer = crypto.createSign('DSA-SHA256');

  signer.write(doc);

  signer.end();

  const signature = signer.sign(private_key, 'hex');

  console.log('Digital Signature: ', signature);

  fs.writeFileSync('signature.hex', signature);

  return signature;

};
```

Додаток Е

Код створення процедури верифікації підпису

```
const crypto = require('crypto');

const fs = require('fs');

module.exports.verify = (filename, isKeyUploaded, isSignatureUploaded) => {

  const publicKeyFolder = isKeyUploaded ? 'uploads/' : 'keys/';

  const signatureFolder = isSignatureUploaded ? 'uploads/' : '';

  let public_key = fs.readFileSync(`${publicKeyFolder}publicKey.pem`, 'utf-8');

  const signature = fs.readFileSync(`${signatureFolder}signature.hex`, 'utf-8');

  const doc = fs.readFileSync('uploads/' + filename);

  const verifier = crypto.createVerify('DSA-SHA256');

  verifier.write(doc);

  verifier.end();

  const result = verifier.verify(public_key, signature, 'hex');

  console.log('Digital Signature Verification : ' + result);

  return result;

};
```