МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РЕАЛІЗАЦІЯ КРИПТОГРАФІЧНОГО ПРОТОКОЛУ ЦИФРОВОГО ПІДПИСУ»

Виконав: здобувач освіти 4 курсу, групи К121-20
Спеціальність 121 Інженерія
програмного забезпечення
(шифр і спеціальність)
Олександр МИКОЛЕНКО
(ім'я та ПРІЗВИЩЕ)
Керівник Анна НЕЛАСА
(ім'я та ПРІЗВИЩЕ)

Рецензент доцент кафедри програмних засобів НУ "Запорізька політехніка", доцент, к.т.н. Олександр СТЕПАНЕНКО
(посада, вчене звання, науковий ступінь, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2024

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо–кваліфікаційний рівень фаховий молодший бакалавр

Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Т.М. Смолянкова
(підпис)

“ 14 ” червня 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

МИКОЛЕНКУ Олександру Андрійовичу
(прізвище, ім’я та по– батькові)

1. Тема роботи «Реалізація криптографічного протоколу цифрового підпису»

Керівник роботи к.т.н. НЕЛАСА Анна Вікторівна
(науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від « 30 » листопада 2024 р. № 2004-с

2. Строк подання студентом роботи

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково– пояснювальної записки (перелік питань, які потрібно розробити)
1.Сучасні інформаційні системи
2. Розробка проекту інформаційної системи
3. Програмна реалізація інформаційної системи

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-3	Анна Неласа		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	Постановка задачі	Грудень 2023	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2024	виконано
3.	Розробка першого розділу:		
	Огляд інформаційних систем та технологій для їхнього створення	Лютий 2024	виконано
4.	Розробка другого розділу:		
	Розробка проєкту інформаційної системи та вибір технологій	Квітень 2024	виконано
5.	Розробка третього розділу:		
	Розробка застосунку	Травень 2024	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2024	виконано
7.	Захист кваліфікаційної роботи	21.06.2024	виконано

Здобувач освіти _____ Олександр МИКОЛЕНКО
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Анна Неласа
(підпис) (ім'я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Кваліфікаційна робота: 47 сторінок, 18 позицій у переліку посилань, 2 діаграми.

Об’єкт дослідження - законодавче регулювання та практичне використання криптографічних протоколів цифрового підпису в Україні.

Предмет дослідження - законодавчі аспекти та технічна реалізація криптографічних протоколів, зокрема алгоритму цифрового підпису на еліптичних кривих (ECDSA) з використанням OpenSSL у C++.

Мета дослідження - аналіз існуючої законодавчої бази України щодо використання цифрового підпису та розробка практичних рекомендацій з його впровадження з використанням сучасних криптографічних протоколів.

У кваліфікаційній роботі викладено теоретичні відомості про криптографічні протоколи цифрового підпису, зокрема алгоритм ECDSA та його реалізацію з використанням OpenSSL у C++. Розглянуто законодавчі акти України, що регулюють використання цифрового підпису, включаючи Закон України "Про електронні довірчі послуги" та Закон України "Про захист інформації в інформаційно-телекомунікаційних системах". Проаналізовано вимоги до кваліфікованого електронного підпису та постачальників електронних довірчих послуг. В практичній частині роботи представлено розробку прикладу реалізації алгоритму ECDSA на основі бібліотеки OpenSSL у середовищі C++.

КРИПТОГРАФІЧНИЙ ПРОТОКОЛ, ЦИФРОВИЙ ПІДПИС, ECDSA, OPENSSL, C++, ЗАКОНОДАВСТВО УКРАЇНИ, ЕЛЕКТРОННІ ДОВІРЧІ ПОСЛУГИ, ЗАХИСТ ІНФОРМАЦІЇ, СЕРТИФІКАЦІЯ КЛЮЧІВ, КВАЛІФІКОВАНИЙ ЕЛЕКТРОННИЙ ПІДПИС, АУДИТ І БЕЗПЕКА

SUMMARY

Diploma thesis: 47 pages, 18 references, 2 diagrams.

The object of research is the legislative regulation and practical use of cryptographic digital signature protocols in Ukraine.

The subject of the study is the legislative aspects and technical implementation of cryptographic protocols, in particular the Elliptic Curve Digital Signature Algorithm (ECDSA) using OpenSSL in C++.

The purpose of the study is to analyze the existing legislative framework of Ukraine regarding the use of digital signatures and to develop practical recommendations for its implementation using modern cryptographic protocols.

The thesis project presents theoretical information about digital signature cryptographic protocols, in particular the ECDSA algorithm and its implementation using OpenSSL in C++. The legislative acts of Ukraine regulating the use of digital signatures, including the Law of Ukraine “On Electronic Trust Services” and the Law of Ukraine “On Protection of Information in Information and Telecommunication Systems” are considered. The author analyzes the requirements for a qualified electronic signature and providers of electronic trust services. The practical part of the paper presents the development of an example of implementation of the ECDSA algorithm based on the OpenSSL library in the C++ environment.

CRYPTOGRAPHIC PROTOCOL, DIGITAL SIGNATURE, ECDSA, OPENSSL, C++, UKRAINIAN LEGISLATION, ELECTRONIC TRUST SERVICES, INFORMATION PROTECTION, KEY CERTIFICATION, QUALIFIED ELECTRONIC SIGNATURE, AUDIT AND SECURITY

ЗМІСТ

ВСТУП

Мета дослідження

Метою даної дипломної роботи є розробка та реалізація криптографічного протоколу цифрового підпису на мові програмування C++, який відповідає сучасним вимогам безпеки і може бути застосований у різних сферах, таких як електронний документообіг, банківські операції та інші області, де важлива цілісність і автентичність інформації.

Теоретичні основи

Цифровий підпис є математичною схемою, яка використовується для перевірки автентичності цифрових повідомлень чи документів. Основою цифрового підпису є криптографічні алгоритми з відкритим ключем, такі як RSA, DSA (Digital Signature Algorithm) та ECDSA (Elliptic Curve Digital Signature Algorithm). Принцип роботи цифрового підпису полягає у створенні унікального підпису за допомогою приватного ключа, який може бути перевірений за допомогою відповідного публічного ключа.

Етапи реалізації

1. Вибір алгоритму цифрового підпису: Для реалізації було обрано алгоритм ECDSA, який забезпечує високий рівень безпеки при відносно невеликому розмірі ключів.

2. Генерація ключів: На першому етапі реалізації створюються пари ключів (приватний і публічний). Приватний ключ використовується для підпису повідомлення, а публічний — для перевірки підпису.

3. Процес підписання: За допомогою приватного ключа та хеш-функції створюється цифровий підпис для даного повідомлення. Хеш-функція перетворює повідомлення у фіксований розмір, що значно спрощує процес підписання.

4. Перевірка підпису: Для перевірки автентичності повідомлення використовується публічний ключ та отриманий підпис. Якщо перевірка

успішна, це підтверджує, що повідомлення не було змінено і було створено власником відповідного приватного ключа.

Практична реалізація

Для практичної реалізації було використано мову програмування C++ з бібліотеками, що підтримують криптографічні операції, такими як OpenSSL. Реалізація включає генерацію ключів, підписання повідомлення та перевірку підпису. Додатково були розроблені тестові сценарії для перевірки коректності роботи протоколу в різних умовах.

1 ЗАКОНОДАВЧЕ РЕГУЛЮВАННЯ КРИПТОГРАФІЧНОГО ПРОТОКОЛУ ЦИФРОВОГО ПІДПИСУ В УКРАЇНІ

1.1 Вступ

Законодавче регулювання криптографічних протоколів, зокрема цифрового підпису, є важливим аспектом забезпечення їх використання в юридично значущих діях. Це включає в себе визначення правових основ, забезпечення відповідності стандартам безпеки, управління ризиками та захистом інформації. У цьому розділі ми детально розглянемо основні закони та нормативні акти України, які регулюють використання цифрових підписів, а також практичні аспекти їх реалізації та інтеграції у різні сфери діяльності.

1.2 Основні нормативно-правові акти

1.2.1 Закон України "Про електронні довірчі послуги"

Закон України "Про електронні довірчі послуги" є основним нормативним актом, що регулює використання електронних підписів в Україні. Він встановлює правові основи для надання електронних довірчих послуг, визначає вимоги до постачальників таких послуг, а також порядок використання електронних підписів у різних сферах діяльності.

Основні положення закону:

- Статус електронного підпису: Визначає, що електронний підпис має таку ж юридичну силу, як і власноручний підпис, за умови дотримання встановлених вимог.

- Кваліфікований електронний підпис: Встановлює вимоги до кваліфікованого електронного підпису, який забезпечує вищий рівень безпеки та автентифікації.

- Постачальники довірчих послуг: Визначає вимоги до постачальників довірчих послуг, їх обов’язки та відповідальність.

1.2.2 Закон України "Про захист інформації в інформаційно-телекомунікаційних системах"

Цей закон регулює питання захисту інформації в інформаційно-телекомунікаційних системах, включаючи використання криптографічних засобів. Він визначає основні вимоги до захисту інформації, порядок проведення експертизи криптографічних засобів та їх сертифікації.

Основні положення закону:

- Вимоги до захисту інформації: Визначає загальні вимоги до захисту інформації, зокрема з використанням криптографічних засобів.

- Експертиза та сертифікація: Встановлює порядок проведення експертизи та сертифікації криптографічних засобів для використання в інформаційно-телекомунікаційних системах.

- Контроль за дотриманням вимог: Визначає механізми контролю за дотриманням вимог щодо захисту інформації.

1.3 Вимоги до використання цифрових підписів

1.3.1 Вимоги до кваліфікованого електронного підпису

Кваліфікований електронний підпис (КЕП) вимагає використання сертифікатів відкритих ключів, які видаються акредитованими центрами сертифікації. Основні вимоги до КЕП включають:

- Сертифікати відкритих ключів: Повинні бути видані акредитованими центрами сертифікації, які відповідають вимогам законодавства.

- Захист приватних ключів: Приватні ключі повинні зберігатися у захищеному середовищі, наприклад, у криптографічних токенах або апаратних модулях безпеки.

- Відповідність стандартам: Алгоритми криптографічного підпису повинні відповідати міжнародним та національним стандартам.

1.3.2 Використання електронних підписів у різних сферах

Електронні підписи можуть використовуватися у різних сферах діяльності, включаючи:

- Електронний документообіг: Забезпечує юридичну значущість електронних документів.

- Електронна комерція: Дозволяє укладати юридично значущі договори та угоди в електронній формі.

- Державні послуги: Використання електронних підписів для взаємодії з державними органами та отримання державних послуг в електронному вигляді.

1.4 Центри сертифікації ключів

1.4.1 Акредитація центрів сертифікації

Акредитація центрів сертифікації здійснюється відповідно до вимог законодавства. Центри сертифікації повинні відповідати певним критеріям, включаючи технічні, організаційні та безпекові вимоги. Акредитація

передбачає перевірку готовності центру забезпечувати надійне зберігання та управління криптографічними ключами.

1.4.2 Функції центрів сертифікації

Центри сертифікації виконують наступні функції:

- Видача сертифікатів: Видача сертифікатів відкритих ключів для кваліфікованих електронних підписів.
- Управління сертифікатами: Забезпечення управління життєвим циклом сертифікатів, включаючи їх видачу, відкликання та оновлення.
- Перевірка сертифікатів: Надання послуг з перевірки дійсності сертифікатів.

1.5 Вимоги до безпеки криптографічних протоколів

1.5.1 Стандарти безпеки

Криптографічні протоколи, зокрема цифровий підпис, повинні відповідати міжнародним та національним стандартам безпеки. В Україні використовуються стандарти, розроблені Державною службою спеціального зв’язку та захисту інформації України, а також міжнародні стандарти, такі як ISO/IEC 15408 (Критерії оцінки безпеки інформаційних технологій) та FIPS 140-2 (Стандарти безпеки криптографічних модулів).

1.5.2 Захист криптографічних ключів

Забезпечення захисту криптографічних ключів є критично важливим аспектом безпеки. Основні вимоги включають:

- Безпечне зберігання: Приватні ключі повинні зберігатися в захищених середовищах, таких як апаратні модулі безпеки (HSM) або криптографічні токени.

- Контроль доступу: Доступ до ключів повинен бути обмеженим та контролюватися за допомогою багатофакторної автентифікації.

- Аудит і моніторинг: Впровадження систем аудиту та моніторингу для виявлення і попередження несанкціонованого доступу до криптографічних ключів.

1.5.3 Вимоги до криптографічних алгоритмів

Використовувані криптографічні алгоритми повинні відповідати сучасним вимогам безпеки та надавати необхідний рівень криптографічного захисту. Для цифрових підписів можуть використовуватися такі алгоритми, як ECDSA (алгоритм цифрового підпису на еліптичних кривих) або RSA (алгоритм асиметричного шифрування).

1.6 Юридичне визнання та використання електронних підписів

1.6.1 Юридична сила електронного підпису

Відповідно до українського законодавства, електронний підпис має таку ж юридичну силу, як і власноручний підпис, за умови дотримання всіх необхідних вимог. Це означає, що документи, підписані кваліфікованим електронним підписом, визнаються юридично значущими і можуть використовуватися у суді як докази.

1.6.2 Використання в електронному документообігу

Електронний документообіг передбачає використання електронних підписів для підписання документів та їх обміну між сторонами. Це значно спрощує процес документообігу, зменшує витрати на папір і зберігання документів, а також підвищує швидкість обробки інформації.

1.6.3. Використання у державних послугах

Державні послуги, що надаються в електронному вигляді, часто вимагають використання електронних підписів для автентифікації користувачів і підписання документів. Це включає послуги з подачі податкових декларацій, отримання адміністративних послуг та реєстрації юридичних осіб.

1.7 Вимоги до постачальників електронних довірчих послуг

1.7.1 Акредитація та контроль

Постачальники електронних довірчих послуг повинні проходити акредитацію відповідно до вимог законодавства. Це включає перевірку їх технічної бази, процесів безпеки та організаційної структури. Акредитація проводиться Державною службою спеціального зв’язку та захисту інформації України.

1.7.2 Обов’язки постачальників

Основні обов’язки постачальників електронних довірчих послуг включають:

- Видача сертифікатів: Надання послуг з видачі сертифікатів відкритих ключів.

- Управління ключами: Забезпечення безпеки та управління життєвим циклом криптографічних ключів.

- Захист даних: Забезпечення захисту даних користувачів та інформації, що обробляється в процесі надання довірчих послуг.

1.8 Практичні аспекти використання електронних підписів в Україні

1.8.1 Електронна комерція

В електронній комерції електронні підписи використовуються для підписання договорів, рахунків-фактур та інших юридично значущих документів. Це дозволяє здійснювати комерційні операції швидше і зручніше, забезпечуючи при цьому високий рівень безпеки.

1.8.2 Фінансовий сектор

У фінансовому секторі електронні підписи застосовуються для підписання фінансових звітів, банківських документів та інших юридично значущих документів. Це сприяє підвищенню ефективності операцій та зниженню ризиків шахрайства.

1.8.3 Державне управління

У державному управлінні електронні підписи використовуються для забезпечення безпеки та автентифікації електронних документів, що

обмінюються між державними органами та громадянами. Це включає такі процеси, як подача податкових декларацій, реєстрація підприємств та отримання адміністративних послуг.

1.9 Вимоги до безпеки криптографічних протоколів

1.9.1 Стандарти безпеки

Криптографічні протоколи, зокрема цифровий підпис, повинні відповідати міжнародним та національним стандартам безпеки. В Україні використовуються стандарти, розроблені Державною службою спеціального зв'язку та захисту інформації України, а також міжнародні стандарти, такі як ISO/IEC 15408 (Критерії оцінки безпеки інформаційних технологій) та FIPS 140-2 (Стандарти безпеки криптографічних модулів).

1.9.2 Захист криптографічних ключів

Забезпечення захисту криптографічних ключів є критично важливим аспектом безпеки. Основні вимоги включають:

- Безпечне зберігання: Приватні ключі повинні зберігатися в захищених середовищах, таких як апаратні модулі безпеки (HSM) або криптографічні токени.

- Контроль доступу: Доступ до ключів повинен бути обмеженим та контролюватися за допомогою багатофакторної автентифікації.

- Аудит і моніторинг: Впровадження систем аудиту та моніторингу для виявлення і попередження несанкціонованого доступу до криптографічних ключів.

1.9.3 Вимоги до криптографічних алгоритмів

Використовувані криптографічні алгоритми повинні відповідати сучасним вимогам безпеки та надавати необхідний рівень криптографічного захисту. Для цифрових підписів можуть використовуватися такі алгоритми, як ECDSA (алгоритм цифрового підпису на еліптичних кривих) або RSA (алгоритм асиметричного шифрування).

1.10 Перспективи розвитку законодавства у сфері криптографічних протоколів

1.10.1 Врахування міжнародного досвіду

Законодавство України у сфері криптографічних протоколів постійно розвивається, враховуючи міжнародний досвід та сучасні технологічні тенденції. Важливо інтегрувати передовий досвід країн Європейського Союзу та інших розвинених держав для забезпечення високого рівня безпеки та надійності.

1.10.2 Адаптація до нових загроз

З розвитком технологій зростають і загрози для інформаційної безпеки. Законодавство має адаптуватися до нових викликів, включаючи загрози квантової криптографії, що можуть значно змінити сучасні підходи до захисту інформації.

1.10.3 Підтримка інновацій

Підтримка інновацій та розвиток нових технологій є ключовим аспектом забезпечення конкурентоспроможності країни у глобальній економіці. Законодавство має створювати сприятливі умови для досліджень і розробок у сфері криптографії та інформаційної безпеки.

2 ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВОГО ПІДПИСУ

2.1 Вступ до цифрового підпису

Цифровий підпис є основним інструментом для забезпечення інформаційної безпеки в сучасному світі. Він підтверджує автентичність, цілісність та невідмовність електронних документів і повідомлень. Використання цифрових підписів є критично важливим для електронного документообігу, банківських операцій, електронної комерції та інших сфер, де потрібна безпека та довіра до інформації.

2.2 Основні поняття та терміни

- Цифровий підпис**:** Криптографічна схема для підтвердження того, що повідомлення або документ було створене власником приватного ключа. Він гарантує автентичність, цілісність та невідмовність.
- Приватний ключ**:** Секретний ключ, використовуваний для створення цифрового підпису. Він повинен бути захищений від несанкціонованого доступу.
- Публічний ключ**:** Відкритий ключ для перевірки цифрового підпису. Він може бути публічно доступний.
- Хеш-функція**:** Математична функція, яка перетворює дані у фіксоване хеш-значення, стійке до колізій.

2.3 Історія та розвиток цифрового підпису

Цифровий підпис був запропонований Уітфілдом Діффі та Мартіном Хеллманом у 1976 році, з введенням криптографії з відкритим ключем. Перший практичний алгоритм цифрового підпису, RSA, був створений у

1978 році Рональдом Рівестом, Аді Шаміром і Леонардом Адлеманом. Відтоді розвиток продовжувався, включаючи алгоритми DSA та ECDSA.

2.4 Математичні основи цифрового підпису

Цифрові підписи базуються на складних математичних задачах. Основні алгоритми включають RSA, DSA та ECDSA.

2.4.1 Алгоритм RSA

RSA використовує складність факторизації великих простих чисел і працює за таким принципом:

1. Генерація ключів: Вибираються два великі прості числа, їх добуток утворює модуль. Експоненти для шифрування і розшифрування вибираються взаємно простими з функцією Ейлера.
2. Створення підпису: Хеш повідомлення шифрується приватним ключем, створюючи унікальний підпис.
3. Перевірка підпису: Підпис розшифровується публічним ключем і порівнюється з хешем повідомлення.

2.4.2 Алгоритм DSA

DSA базується на дискретному логарифмуванні в кінцевих полях і використовує параметри p, q, g.

1. Генерація ключів: Вибираються p, q, g.
2. Створення підпису: Хеш повідомлення підписується приватним ключем за допомогою параметрів p, q, g.
3. Перевірка підпису: Використовується публічний ключ та параметри p, q, g.

2.4.3 Алгоритм ECDSA

ECDSA використовує еліптичні криві для забезпечення безпеки.

1. Генерація ключів: Вибирається еліптична крива, створюється пара ключів.
2. Створення підпису: Хеш повідомлення підписується приватним ключем.
3. Перевірка підпису: Використовується публічний ключ та параметри кривої.

2.5 Сучасні стандарти та протоколи цифрового підпису

Існує декілька стандартів, що регулюють використання цифрових підписів:

- PKCS 1**:** Стандартизація використання RSA.
- FIPS 186-4**:** Стандартизація алгоритмів DSA і ECDSA.
- X.509**:** Стандарт для PKI та сертифікатів.

2.6 Застосування цифрового підпису

Цифрові підписи широко використовуються:

- Електронний документообіг**:** Забезпечують юридичну силу електронних документів.
- Банківські операції**:** Гарантують безпеку фінансових транзакцій.
- Електронна комерція**:** Захищають дані під час електронних платежів.
- Інформаційні системи**:** Захищають дані від несанкціонованого доступу.

2.7 Проблеми та виклики

Цифрові підписи стикаються з такими викликами:

- Криптографічна стійкість: Постійне вдосконалення алгоритмів для протидії новим загрозам.
- Захист приватного ключа**:** Захист ключів від несанкціонованого доступу.
- Правові аспекти: Уніфікація та гармонізація законодавчих норм.

2.8 Перспективи розвитку

Розвиток технологій цифрового підпису включає:

- Покращення алгоритмів**:** Створення ефективніших і безпечніших алгоритмів.
- Квантова криптографія: Захист від квантових комп’ютерів.
- Інтеграція з новими технологіями**:** Використання в блокчейні, IoT та інших технологіях.

2.9 Технологічні та архітектурні аспекти впровадження цифрових підписів

2.9.1 Інфраструктура відкритих ключів (PKI)

PKI складається з:

- Центр сертифікації (CA)**:** Видає сертифікати.
- Реєстраційний центр (RA)**:** Ідентифікує користувачів перед видачею сертифікатів.

- Сховище сертифікатів**:** Зберігає сертифікати.
- Механізми управління ключами**:** Генерація, зберігання, розподіл і відкликання ключів.

2.9.2 Протоколи та стандарти

Протоколи, що забезпечують взаємодію PKI:

- S/MIME**:** Захищений обмін електронною поштою.
- TLS: Безпека передачі даних в Інтернеті.
- XMLDSig: Підпис XML-документів.
- PDF Signatures: Підпис PDF-документів.

2.10 Практичні аспекти впровадження цифрових підписів

2.10.1 Вибір алгоритмів та параметрів

При впровадженні цифрових підписів важливо правильно вибрати алгоритми і параметри, які забезпечують необхідний рівень безпеки. Основні критерії вибору включають:

Безпека: Алгоритм повинен бути стійким до відомих атак і забезпечувати необхідний рівень криптографічного захисту.

Ефективність: Алгоритм повинен бути ефективним з точки зору обчислювальних ресурсів і часу виконання, особливо для обмежених пристроїв.

Сумісність: Алгоритм повинен бути сумісним з існуючими стандартами і протоколами, щоб забезпечити інтероперабельність з іншими системами.

2.10.2 Управління ключами та сертифікатами

Управління сертифікатами та ключами є одним з основни

Генерація ключів: Ключі повинні генеруватися у безпечному середовищі з використанням криптографічно стійких методів.

Зберігання ключів: Приватні ключі повинні зберігатися у захищеному середовищі, наприклад, у апаратних модулях безпеки (HSM) або захищених програмних контейнерах.

Розподіл ключів: Ключі повинні безпечно передаватися між користувачами та системами, з використанням захищених каналів зв’язку.

Відкликання сертифікатів: У випадку компрометації ключів або зміни статусу користувача сертифікати повинні бути оперативно відкликані і включені у списки відкликаних сертифікатів (CRL).

2.10.3 Правові та регуляторні аспекти

Впровадження цифрових підписів повинно враховувати правові та регуляторні вимоги, які можуть відрізнятися в різних юрисдикціях. Основні аспекти включають:

Юридична сила: Цифрові підписи повинні мати юридичну силу і визнаватися судами та іншими правовими інстанціями. Це вимагає відповідності нормативним актам і стандартам, таким як eIDAS в Європейському Союзі або Закон про електронні підписи у США. Захист даних: Використання цифрових підписів повинно відповідати вимогам захисту персональних даних, наприклад, GDPR в Європейському Союзі. Це включає забезпечення конфіденційності, цілісності та доступності даних. Аудит і відповідність: Організації повинні проводити регулярні аудити своїх систем цифрових підписів, щоб забезпечити відповідність нормативним вимогам і стандартам безпеки. Аудити можуть включати перевірку процесів генерації і зберігання ключів, управління сертифікатами, а також заходів захисту даних.

3 АЛГОРИТМ ECDSA ТА ВИКОРИСТАННЯ OPENSSL У C++

3.1 Вступ до ECDSA

Алгоритм цифрового підпису на основі еліптичних кривих (ECDSA) є одним із найпопулярніших алгоритмів у сучасній криптографії. Його основна перевага полягає в забезпеченні високого рівня безпеки при відносно невеликій довжині ключів у порівнянні з іншими алгоритмами, такими як RSA.

3.2 Основи алгоритму ECDSA

ECDSA використовує математичні властивості еліптичних кривих над кінцевими полями для створення та перевірки цифрових підписів. Основні компоненти алгоритму включають:

3.2.1 Еліптичні криві

Еліптична крива визначається рівнянням виду:

$y^2 = x^3 + ax + b$

де a і b є константами, які визначають форму кривої. Для використання в криптографії ці криві мають спеціальні властивості, що забезпечують високий рівень безпеки.

3.2.2 Генерація ключів

Генерація ключів у ECDSA включає вибір приватного та публічного ключів. Приватний ключ d є випадковим числом, а публічний ключ Q є точкою на кривій, отриманою шляхом множення генератора кривої G на приватний ключ: $Q = d * G$.

3.2.3 Створення підпису

Процес створення підпису включає наступні кроки:

1. Обчислення хеш-значення повідомлення $e = HASH(m)$.
2. Вибір випадкового числа k у діапазоні [1, n-1], де n - порядок точки G.
3. Обчислення точки $R = k * G$ і взяття координати x цієї точки $r = x_R \bmod n$.
4. Обчислення значення $s = k^{-1} (e + d * r) \bmod n$.

Пара (r, s) є цифровим підписом повідомлення.

3.2.4 Перевірка підпису

Процес перевірки підпису включає наступні кроки:

1. Обчислення хеш-значення повідомлення $e = HASH(m)$.
2. Обчислення значень

$w = s^{-1} \bmod n$, $u_1 = e * w \bmod n$ та $u_2 = r * w \bmod n$.

3. Обчислення точки $P = u_1 * G + u_2 * Q$.
4. Перевірка рівності $r \equiv x_P \bmod n$, де x_P - координата x точки P.

Якщо рівність виконується, підпис вважається дійсним.

3.3 Використання OpenSSL для реалізації ECDSA у C++

OpenSSL - це потужна бібліотека з відкритим вихідним кодом, яка забезпечує реалізацію багатьох криптографічних алгоритмів, включаючи ECDSA. Вона широко використовується для забезпечення безпеки в інтернет-протоколах і додатках.

3.3.1 Встановлення та налаштування OpenSSL

Для роботи з OpenSSL спочатку потрібно встановити цю бібліотеку. У більшості систем на базі Unix можна скористатися пакетним менеджером, наприклад: sudo apt-get install openssl libssl-dev

Або для систем на базі Red Hat: sudo yum install openssl-devel

3.3.2 Генерація ключів ECDSA за допомогою OpenSSL

Генерацію ключів можна здійснити за допомогою командного рядка OpenSSL, а також програмно за допомогою API OpenSSL у C++. Для генерації ключів за допомогою командного рядка використовуйте такі команди:

```
openssl ecparam -genkey -name secp256k1 -out private_key.pem
openssl ec -in private_key.pem -pubout -out public_key.pem
```

3.3.3 Створення цифрового підпису у C++

Для створення цифрового підпису в C++ за допомогою OpenSSL, можна скористатися наступним кодом:

```
#include <openssl/evp.h>
#include <openssl/ec.h>
#include <openssl/ecdsa.h>
#include <openssl/pem.h>
#include <fstream>
#include <vector>
#include <iostream>

std::vector<unsigned char> hashMessage(const std::string& message) {
    EVP_MD_CTX* mdctx = EVP_MD_CTX_new();
    const EVP_MD* md = EVP_sha256();
    std::vector<unsigned char> digest(EVP_MD_size(md));

    EVP_DigestInit_ex(mdctx, md, nullptr);
    EVP_DigestUpdate(mdctx, message.c_str(), message.size());
    unsigned int digest_len;
    EVP_DigestFinal_ex(mdctx, digest.data(), &digest_len);
    EVP_MD_CTX_free(mdctx);

    digest.resize(digest_len);  // Adjust the size to the actual length of the digest
    return digest;
}

std::vector<unsigned char> signMessage(EVP_PKEY* pkey, const std::vector<unsigned char>&
hash) {
    EVP_MD_CTX* mdctx = EVP_MD_CTX_new();
    EVP_PKEY_CTX* pctx = nullptr;
    std::vector<unsigned char> signature(EVP_PKEY_get_size(pkey));
    size_t siglen = signature.size();

    EVP_DigestSignInit(mdctx, &pctx, EVP_sha256(), nullptr, pkey);
    EVP_DigestSign(mdctx, signature.data(), &siglen, hash.data(), hash.size());
    signature.resize(siglen);

    EVP_MD_CTX_free(mdctx);
    return signature;
}

int main() {
    // Завантаження приватного ключа
    const char* privateKeyPath = "private_key.pem";
    FILE* fp = fopen(privateKeyPath, "r");
    if (!fp) {
```

```
        std::cerr << "Помилка відкриття файлу приватного ключа: " << privateKeyPath <<
std::endl;
        perror("Причина");
        return 1;
    }

    EVP_PKEY* pkey = PEM_read_PrivateKey(fp, nullptr, nullptr, nullptr);
    fclose(fp);

    if (!pkey) {
        std::cerr << "Помилка завантаження приватного ключа" << std::endl;
        return 1;
    }

    // Повідомлення для підпису
    std::string message = "Hello, ECDSA with OpenSSL in C++!";
    std::vector<unsigned char> hash = hashMessage(message);
    std::vector<unsigned char> signature = signMessage(pkey, hash);

    // Збереження підпису у файл
    std::ofstream sigFile("signature.bin", std::ios::binary);
    sigFile.write(reinterpret_cast<const char*>(signature.data()), signature.size());
    sigFile.close();

    EVP_PKEY_free(pkey);
    return 0;
}
```

Рисунок 3.1 – Створення цифрового підпису

3.3.4 Перевірка цифрового підпису у C++

Для перевірки цифрового підпису у C++ за допомогою OpenSSL, використовуйте наступний код:

```
#include <openssl/evp.h>
#include <openssl/ec.h>
#include <openssl/ecdsa.h>
#include <openssl/sha.h>
#include <openssl/pem.h>
#include <fstream>
#include <vector>
#include <iostream>

// Функція для хешування повідомлення
std::vector<unsigned char> hashMessage(const std::string &message) {
    std::vector<unsigned char> hash(SHA256_DIGEST_LENGTH);
    SHA256(reinterpret_cast<const unsigned char*>(message.data()), message.size(),
hash.data());
    return hash;
}

bool verifySignature(EVP_PKEY *pkey, const std::vector<unsigned char> &hash, const
std::vector<unsigned char> &signature) {
    EVP_MD_CTX *mdctx = EVP_MD_CTX_new();
    EVP_PKEY_CTX *pctx = nullptr;

    EVP_DigestVerifyInit(mdctx, &pctx, EVP_sha256(), nullptr, pkey);
    bool result = EVP_DigestVerify(mdctx, signature.data(), signature.size(), hash.data(),
hash.size()) == 1;

    EVP_MD_CTX_free(mdctx);
    return result;
}

int main() {
    // Завантаження публічного ключа
    FILE *fp = fopen("public_key.pem", "r");
    if (!fp) {
        std::cerr << "Помилка відкриття файлу з публічним ключем" << std::endl;
        return 1;
    }
    EVP_PKEY *pkey = PEM_read_PUBKEY(fp, nullptr, nullptr, nullptr);
    fclose(fp);

    if (!pkey) {
        std::cerr << "Помилка завантаження публічного ключа" << std::endl;
        return 1;
```

```
    }

    // Повідомлення для перевірки
    std::string message = "Hello, ECDSA with OpenSSL in C++!";
    std::vector<unsigned char> hash = hashMessage(message);

    // Завантаження підпису з файлу
    std::ifstream sigFile("signature.bin", std::ios::binary);
    if (!sigFile) {
        std::cerr << "Помилка відкриття файлу з підписом" << std::endl;
        EVP_PKEY_free(pkey);
        return 1;
    }
    std::vector<unsigned char> signature((std::istreambuf_iterator<char>(sigFile)),
std::istreambuf_iterator<char>());
    sigFile.close();

    // Перевірка підпису
    if (verifySignature(pkey, hash, signature)) {
        std::cout << "Підпис дійсний" << std::endl;
    } else {
        std::cout << "Підпис недійсний" << std::endl;
    }

    EVP_PKEY_free(pkey);
    return 0;
}
```

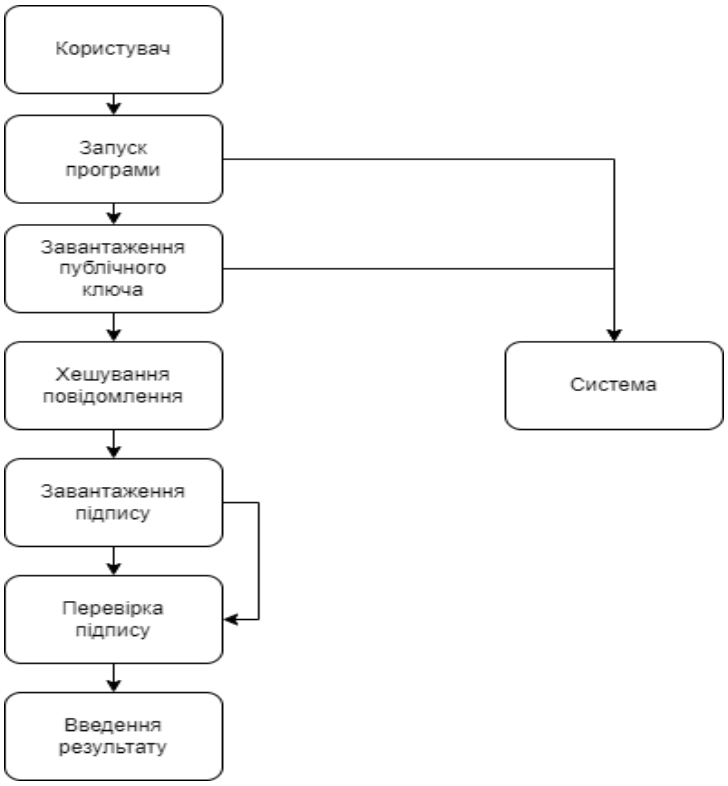

Рисунок 3.2 - Перевірка цифрового підпису

3.4 Переваги та недоліки ECDSA

3.4.1 Переваги

1. Ефективність**:** ECDSA забезпечує високий рівень безпеки при меншій довжині ключів у порівнянні з RSA. Це дозволяє зменшити обчислювальні витрати та обсяг переданих даних.

2. Безпека: Складність алгоритму базується на проблемі дискретного логарифмування на еліптичних кривих, що забезпечує високий рівень криптографічного захисту.

3. Компактність: Ключі та підписи ECDSA значно менші за розміром у порівнянні з іншими алгоритмами, що робить його ідеальним для використання на пристроях з обмеженими ресурсами.

3.4.2 Недоліки

1. Складність реалізації**:** Реалізація алгоритмів на еліптичних кривих вимагає глибоких знань у галузі криптографії та математики.

2. Патентні обмеження**:** У минулому використання деяких еліптичних кривих було обмежене патентами, хоча більшість із них вже втратили свою чинність.

3.5 Практичні аспекти використання ECDSA з OpenSSL у C++

3.5.1 Інтеграція з програмним забезпеченням

OpenSSL дозволяє легко інтегрувати функціональність ECDSA в різні програмні продукти. Для цього можна використовувати бібліотеки та API, які забезпечують роботу з криптографічними операціями на еліптичних кривих.

3.5.2 Безпека ключів

Забезпечення безпеки приватних ключів є критично важливим аспектом. Рекомендується використовувати апаратні модулі безпеки (HSM) для зберігання та управління ключами. Крім того, варто застосовувати методи контролю доступу та регулярні аудити безпеки.

3.5.3 Управління сертифікатами

ECDSA ключі можуть використовуватися в сертифікатах, виданих центрами сертифікації (CA). Управління сертифікатами включає генерацію запитів на сертифікати, їх обробку та регулярне оновлення.

3.6 Використання ECDSA в реальних додатках

ECDSA широко використовується в різних додатках та протоколах, включаючи:

- TLS/SSL**:** Протоколи безпеки для забезпечення захищеного з'єднання між клієнтами і серверами. Використання ECDSA зменшує навантаження на сервери та покращує продуктивність.
- Блокчейн**:** Технології блокчейн, такі як Bitcoin, використовують ECDSA для забезпечення безпеки транзакцій. Це дозволяє гарантувати автентичність і цілісність даних.
- Електронні паспорти**:** ECDSA використовується для захисту даних у біометричних паспортах, що забезпечує високу ступінь безпеки та автентичності даних.

3.7 Алгоритми, що використовуються з ECDSA

3.7.1 Хеш-алгоритми

ECDSA використовує хеш-алгоритми для обчислення хеш-значень повідомлень, які підписуються. Найпоширенішими хеш-алгоритмами є SHA-256, SHA-384 і SHA-512. Вибір хеш-алгоритму залежить від рівня безпеки, що необхідний для конкретного додатка.

3.7.2 Генерація випадкових чисел

Процес створення підпису в ECDSA вимагає генерації криптографічно стійких випадкових чисел. Використання недетермінованих або слабких генераторів випадкових чисел може призвести до серйозних вразливостей.

3.7.3 Бібліотеки криптографії

Для реалізації ECDSA на практиці часто використовуються бібліотеки, такі як OpenSSL, Bouncy Castle, і Crypto++. Ці бібліотеки надають готові до використання функції для генерації ключів, підписання і перевірки підписів.

3.8 Оптимізація продуктивності ECDSA

Оскільки ECDSA часто використовується в ресурсомістких додатках, таких як блокчейн та мобільні пристрої, оптимізація продуктивності є важливим аспектом.

3.8.1 Вибір еліптичних кривих

Вибір правильної еліптичної кривої може значно вплинути на продуктивність. Наприклад, криві з нижчим порядком можуть забезпечувати кращу продуктивність, але знижувати рівень безпеки. Найпоширеніші криві, які використовуються в практичних додатках, це secp256k1 і secp384r1.

3.8.2 Апаратні прискорювачі

Використання апаратних прискорювачів, таких як TPM (Trusted Platform Module) або HSM (Hardware Security Module), може значно підвищити швидкість криптографічних операцій.

3.8.3 Паралельні обчислення

Паралелізація операцій, особливо в контексті багатопоточних середовищ, може забезпечити значне підвищення продуктивності. Багато сучасних бібліотек підтримують паралельні обчислення для підвищення ефективності.

3.9 Використання ECDSA у мобільних додатках

Мобільні додатки часто вимагають високого рівня безпеки при обмежених ресурсах. Використання ECDSA у таких додатках може забезпечити необхідний баланс між безпекою та продуктивністю.

3.9.1 Адаптація до мобільних платформ

На мобільних платформах, таких як iOS та Android, можна використовувати вбудовані криптографічні API для реалізації ECDSA. Наприклад, на iOS доступна бібліотека Security, а на Android - Keystore.

3.9.2 Енергоефективність

Оскільки мобільні пристрої працюють на батареях, енергоефективність криптографічних операцій є критично важливою. Використання оптимізованих алгоритмів та апаратних прискорювачів може значно знизити споживання енергії.

3.10 Використання ECDSA у блокчейні

Блокчейн-технології, такі як Bitcoin та Ethereum, широко використовують ECDSA для забезпечення безпеки транзакцій.

3.10.1 Захист транзакцій

ECDSA використовується для підписання транзакцій, що дозволяє забезпечити їх автентичність і цілісність. Кожна транзакція підписується приватним ключем відправника, що гарантує, що тільки власник цього ключа міг ініціювати транзакцію.

3.10.2 Перевірка транзакцій

Валідація транзакцій у блокчейні включає перевірку підписів, щоб гарантувати, що транзакції не були змінені та що вони дійсно створені власниками відповідних приватних ключів.

3.10.3 Мультипідписи

Деякі блокчейн-платформи підтримують мультипідписи, де транзакція вимагає підписів від кількох користувачів для підтвердження. Це забезпечує додатковий рівень безпеки, знижуючи ризик шахрайства.

3.11 Правові аспекти використання ECDSA

Правові вимоги щодо використання цифрових підписів можуть відрізнятися залежно від юрисдикції. У багатьох країнах існують закони та регуляції, що регулюють використання цифрових підписів.

3.11.1 Визнання цифрових підписів

У багатьох країнах цифрові підписи, включаючи підписи на основі ECDSA, визнані юридично дійсними. Наприклад, у Європейському Союзі діє регламент eIDAS, який забезпечує правову базу для використання цифрових підписів.

3.11.2 Вимоги до безпеки

Юридичні вимоги можуть включати конкретні стандарти безпеки для використання цифрових підписів. Наприклад, може вимагатися використання сертифікатів від довірених центрів сертифікації або дотримання певних криптографічних стандартів.

3.12 Виклики та майбутнє ECDSA

3.12.1 Квантова криптографія

З розвитком квантових обчислень існує потенційна загроза для багатьох сучасних криптографічних алгоритмів, включаючи ECDSA.

Квантові комп'ютери можуть вирішувати певні математичні задачі набагато швидше, ніж класичні комп'ютери, що може компрометувати безпеку ECDSA.

3.12.2 Постквантова криптографія

Розробка постквантових криптографічних алгоритмів, що стійкі до квантових атак, є однією з ключових напрямків розвитку криптографії. Це включає як нові алгоритми, так і модифікації існуючих алгоритмів для забезпечення їх стійкості до квантових атак.

3.12.3 Впровадження стандартів

З часом можуть з'явитися нові стандарти для цифрових підписів, які враховуватимуть як сучасні, так і майбутні загрози. Це включає розробку нових криптографічних протоколів та рекомендацій щодо їх використання.

3.13 Практичні аспекти розгортання ECDSA у великих системах

3.13.1 Інтеграція з існуючими системами

Інтеграція ECDSA у великі системи може вимагати значних змін в архітектурі програмного забезпечення. Важливо забезпечити сумісність між новими криптографічними протоколами та вже існуючими системами автентифікації та авторизації.

3.13.2 Масштабування

Масштабування системи, що використовує ECDSA, є важливим аспектом, особливо в контексті високонавантажених додатків. Необхідно враховувати оптимізацію обчислювальних ресурсів і ефективне управління ключами.

3.13.3 Моніторинг і аудит

Для забезпечення безпеки системи з ECDSA важливо впровадити процеси моніторингу та аудиту. Це включає відстеження спроб несанкціонованого доступу, регулярні перевірки ключів і підписів, а також контроль за оновленнями криптографічних бібліотек.

3.14 Переваги та недоліки використання ECDSA в різних галузях

3.14.1 Фінансові технології

У фінансовій галузі ECDSA забезпечує високий рівень безпеки для електронних транзакцій, захисту даних клієнтів і автентифікації користувачів. Проте складність реалізації та необхідність регулярного оновлення ключів можуть стати викликами.

3.14.2 Інтернет речей (IoT)

Для пристроїв IoT ECDSA є привабливим через свою компактність та ефективність. Проте обмежені ресурси цих пристроїв можуть створювати труднощі в реалізації криптографічних операцій.

3.14.3 Охорона здоров’я

У сфері охорони здоров'я ECDSA може використовуватися для захисту медичних даних і забезпечення конфіденційності пацієнтів. Водночас, необхідність забезпечення сумісності з існуючими системами електронного медичного запису може ускладнювати впровадження.

3.15 Перспективи розвитку та нові напрямки досліджень

3.15.1 Постквантова криптографія

Оскільки квантові обчислення загрожують багатьом сучасним криптографічним алгоритмам, розвиток постквантової криптографії є ключовим напрямком досліджень. Це включає розробку нових алгоритмів, що забезпечують стійкість до квантових атак.

3.15.2 Інтеграція з блокчейн-технологіями

Подальша інтеграція ECDSA з блокчейн-технологіями може сприяти створенню більш безпечних та ефективних децентралізованих систем. Це

включає як покращення існуючих протоколів, так і розробку нових методів захисту даних.

3.15.3 Автоматизація та штучний інтелект

Використання штучного інтелекту для автоматизації процесів, пов'язаних з управлінням ключами і моніторингом безпеки, може значно підвищити ефективність та надійність криптографічних систем.

ВИСНОВКИ

У даній кваліфікаційній роботі було розглянуто та проаналізовано різні аспекти впровадження криптографічного протоколу цифрового підпису в Україні, зокрема алгоритму цифрового підпису на еліптичних кривих (ECDSA) з використанням бібліотеки OpenSSL у C++.

Проведене дослідження дозволило зробити наступні висновки:

1. Аналіз законодавчої бази:

- Українське законодавство чітко регулює використання цифрових підписів через Закони України "Про електронні довірчі послуги" та "Про захист інформації в інформаційно-телекомунікаційних системах". Ці закони забезпечують правову основу для впровадження та використання цифрових підписів у державних та приватних секторах.

- Регуляторні документи та постанови Кабінету Міністрів України визначають порядок акредитації центрів сертифікації ключів та вимоги до постачальників електронних довірчих послуг.

2. Технічні аспекти реалізації цифрового підпису:

- Алгоритм ECDSA є одним з найбільш ефективних та безпечних криптографічних протоколів для створення цифрових підписів. Його використання надає високу ступінь безпеки при відносно невеликій обчислювальній складності.

- Бібліотека OpenSSL є потужним інструментом для реалізації криптографічних алгоритмів, включаючи ECDSA, у середовищі програмування C++. Вона забезпечує необхідний набір функцій для генерації ключів, підписання даних та верифікації підписів.

3. Практична реалізація:

- Розроблено приклад програмного забезпечення для генерації та перевірки цифрових підписів з використанням алгоритму ECDSA та

бібліотеки OpenSSL у середовищі C++. Цей приклад демонструє можливість ефективної інтеграції криптографічних функцій у програмні рішення.

- Використання C++ у поєднанні з OpenSSL дозволяє створювати високопродуктивні та безпечні програмні продукти, які відповідають сучасним вимогам до захисту інформації.

У підсумку, проведене дослідження підтвердило доцільність використання криптографічного протоколу цифрового підпису ECDSA з використанням OpenSSL у C++ в умовах сучасного розвитку інформаційних технологій та потреб у захисті інформації в Україні.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України "Про електронні довірчі послуги". URL: https://zakon.rada.gov.ua/laws/show/2155-19 (дата звернення: 15.06.2024).

2. Закон України "Про захист інформації в інформаційно-телекомунікаційних системах". URL: https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80 (дата звернення: 15.06.2024).

3. Постанова Кабінету Міністрів України "Про затвердження Порядку акредитації центрів сертифікації ключів". URL: https://zakon.rada.gov.ua/laws/show/984-2019-%D0%BF (дата звернення: 15.06.2024).

4. ISO/IEC 15408: "Критерії оцінки безпеки інформаційних технологій". URL: https://www.iso.org/standard/50341.html (дата звернення: 15.06.2024).

5. FIPS 140-2: "Стандарти безпеки криптографічних модулів". URL: https://csrc.nist.gov/publications/detail/fips/140/2/final (дата звернення: 15.06.2024).

6. ECDSA: "Алгоритм цифрового підпису на еліптичних кривих" – технічна документація/ URL: https://datatracker.ietf.org/doc/html/rfc4492 (дата звернення: 15.06.2024).

7. OpenSSL: "Документація по використанню бібліотеки OpenSSL". URL: https://www.openssl.org/docs/ (дата звернення: 15.06.2024).

8. Державна служба спеціального зв’язку та захисту інформації України: "Методичні рекомендації щодо використання криптографічних протоколів". URL: https://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?art (дата звернення: 15.06.2024).

9. Державна служба спеціального зв’язку та захисту інформації України: "Порядок видачі та використання електронних цифрових підписів".

URL: https://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?art_id=61363&c (дата звернення: 15.06.2024).

10. Статті наукових журналів про використання цифрових підписів в електронному урядуванні. URL: https://www.researchgate.net/publication/328518273_The_Use_of_Digital_Signatures_in_E-Government_Services (дата звернення: 15.06.2024).

11. "The Handbook of Applied Cryptography" – Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. URL: https://books.google.com/books/about/Handbook_of_Applied_Cryptography.html?id=rRwnK5Ej-FUC (дата звернення: 15.06.2024).

12. "Cryptography and Network Security: Principles and Practice" – William Stallings. URL: https://www.pearson.com/store/p/cryptography-and-network-security-principles-and-practice/P100000668863 (дата звернення: 15.06.2024).

13. Статті конференцій з криптографії та інформаційної безпеки. URL: https://ieeexplore.ieee.org/Xplore/home.jsp (дата звернення: 15.06.2024).

14. "Practical Cryptography" – Niels Ferguson, Bruce Schneier. URL: https://www.wiley.com/en-us/Practical+Cryptography-p-9781119096726 (дата звернення: 15.06.2024).

15. Документація C++ для роботи з криптографічними бібліотеками. URL: http://www.cplusplus.com/reference/ (дата звернення: 15.06.2024).

16. Рекомендації міжнародних організацій з кібербезпеки щодо використання криптографічних протоколів. URL: https://www.enisa.europa.eu/ (дата звернення: 15.06.2024).

17. Практичні керівництва з реалізації алгоритмів цифрових підписів. URL: https://developer.mozilla.org/en-US/docs/Security/Subresource_Integrity (дата звернення: 15.06.2024).

18. Веб-ресурси та онлайн-платформи з обміну досвідом з криптографії та інформаційної безпеки. URL: https://stackoverflow.com/questions/cryptography (дата звернення: 15.06.2024).

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничого фахового коледжу ЗНУ»

Я, Миколенко Олександр Андрійович, здобувач(-ка) освіти 4 курсу, спеціальності/освітньо-професійної програми Інженерія програмного забезпечення, групи К121-20, адреса електронної пошти epfk121.20mikolenko@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Реалізація криптографічного протоколу цифрового підпису» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата _____ Підпис _____ Олександр МИКОЛЕНКО

Дата 18.06.2024 Підпис _____ Анна НЕЛАСА