МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

УДК 004.421

До захисту допускається
В.о. завідувача кафедри
комп'ютерних наук та інженерії
д.т.н., проф. Рязанцев О. І.
_____
«____»_____ 2021 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

«Дослідження програмних засобів створення цифрового електронного підпису»

Освітньо-кваліфікаційний рівень «Магістр»
Спеціальність 122 «Комп'ютерні науки»

Науковий керівник роботи: _____ Кардашук В. С.
(підпис) (ініціали, прізвище)

Консультант з охорони праці: _____ Критська Я. О.
(підпис) (ініціали, прізвище)

Студент: _____ Сідєльніков В. В.
(підпис) (ініціали, прізвище)

Група: КН-19 дм

Сєвєродонецьк – 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень магістр
Спеціальність 122 «Комп'ютерні науки»

«**ЗАТВЕРДЖУЮ**»
Т.в.о. завідувача кафедри
комп'ютерних наук та інженерії
к.т.н., доц. Кардашук В. С.

"____" _____ 2020 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Сідєльнікову Владиславу Валерійовичу

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи): «Дослідження програмних засобів створення цифрового електронного підпису» затверджена наказом по університету № 136/15.15 від «11» жовтня 2020 р.

2. Строк здачі студентом закінченого проекту (роботи): 10.01.2021 р.

3. Вихідні дані проекту (роботи): матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити):

1. Дослідження алгоритмів створення електронного цифрового підпису.
2. Дослідження існуючих програмних концепцій в області шифрування даних.
3. Дослідження тестування чисел на простоту і вибір параметрів RSA.
4. Алгоритми та моделі криптосистеми з відкритими ключами.
5. Розроблення прикладного модуля створення ЕЦП.
6. Тестування програмних засобів створення ЕЦП та шифрування
6. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точною назвою обов'язкових креслень):

електронні плакати

6. Консультанти роботи, з вказівкою розділів, що до них відносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основна частина	Кардашук В. С.		
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я. О.		

7. Дата видачі завдання: 11.10.2020 р.

Керівник _____ Кардашук В. С.

(підпис)

Завдання до виконання прийняв _____ Сідєльніков В. В.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітки
1.	Отримання завдання, збір матеріалів	11.10.20- 24.10.20	
2.	Огляд програмних засобів створення електронного цифрового підпису	25.10. 20–28.10.20	
3.	Алгоритми та моделі криптосистеми з відкритими ключами	29.10.20– 28.11.20	
4.	Дослідження тестування чисел на простоту і вибір параметрів RSA	28.11.20–31.12.20	
5.	Тестування програмних засобів створення ЕЦП та шифрування	03.01.21 – 04.01.21	
6.	Оформлення пояснювальної записки	05.01.21 – 08.01.21	
7.	Підготовка та подання магістерської роботи до захисту	09.01.21 – 10.01.21	

Студент _____

(підпис)

Науковий керівник _____

(підпис)

АНОТАЦІЯ

Сідєльніков В. В. Дослідження програмних засобів створення цифрового електронного підпису.

Досліджено сучасні алгоритми електронного цифрового підпису (ЕЦП). Алгоритми проаналізовані з точки зору зручності програмної реалізації та швидкодії на різних етапах ЕЦП. Досліджено найбільш застосовувані криптографічні хеш-функції як складова частина системи ЕЦП. Дослідження проведене шляхом програмної реалізації різних систем ЕЦП на мові Java с використанням бібліотеки Open SSL та подальшим програмним профілюванням.

Ключові слова: електронний цифровий підпис, хеш-функція, стандарти цифрового підпису, асиметричні криптографічні системи, швидкодія.

АННОТАЦИЯ

Сидельников В. В. Исследование программных средств создания электронной цифровой подписи.

Исследованы современные алгоритмы электронной цифровой подписи (ЭЦП). Алгоритмы проанализированы с точки зрения удобства программной реализации и быстродействия на различных этапах ЭЦП. Исследованы наиболее применяемые криптографические хэш-функции как составная часть системы ЭЦП. Исследование проведено путем программной реализации различных систем ЭЦП на языке Java с использованием библиотеки Open SSL и последующим программным профилированием.

Ключевые слова: электронная цифровая подпись, хэш-функция, стандарты цифровой подписи, асимметричные криптографические системы, быстродействие

THE ABSTRACT

Sidelnikov V. V. Research of modern digital signature systems.

Modern digital signature (DS) systems were researched. The DS algorithms were analyzed with respect to easiness of software implementation and speed related to their different stages. Common cryptographic hash functions were researched as a part of a DS system.

The research was performed by means of Java software implementation of different DS systems with the use of Open SSL toolkit and further software profiling.

Keywords: digital signature, hash function, digital signature standards, asymmetric cryptographic systems, algorithm speed.

ЗМІСТ

ВСТУП

Електронний цифровий підпис (ЕЦП) (англ. Digital signature) за майже сорокарічну історію свого існування пройшов стрімку еволюцію від математичної ідеї У. Діффі і М. Хеллмана, висловленої у 1976 р. [1], до невід'ємного елементу сучасного захищеного мережевого електронного документообігу. ЕЦП - реквізит електронного документа, призначений для захисту електронного документа від підробки або внесення змін, отриманий в результаті криптографічного перетворення інформації з використанням секретного ключа підпису, що дозволяє ідентифікувати власника ключа підпису і встановити відсутність спотворення інформації в електронному документі.

ЕЦП також забезпечує неможливість відмови особи, яка підписала документ від його акту підписання. Завдяки цим властивостям ЕЦП широко застосовується в наступних сферах:

– безпечний банківський фінансовий оборот,
– юридично значимий електронний документообіг,
– юридична і фінансова обов'язкова звітність перед державними
– органами,
– митне декларування товарів і послуг,
– розрахункові та трейдингові системи,
– дистанційні торгові угоди.

Ухвалення рішень у всіх сферах життєдіяльності підприємства або організації все більшою мірою базується на інформаційних процесах. Аналіз цих процесів з подальшим виробленням управляючих рішень здійснюється на основі інформаційних моделей, побудованих на сучасних інформаційно-телекомунікаційних технологіях. Тому, захист інформації являє собою самостійну складову безпеки підприємства в цілому, значення якої з кожним роком зростає.

Інформаційний ресурс стає одним з головних джерел економічної ефективності підприємства. Фактично спостерігається тенденція, коли всі сфери життєдіяльності підприємства стають залежними від інформаційного розвитку, в процесі якого вони самі породжують інформацію і самі ж її споживають.

На сучасному етапі розвитку основними загрозами безпеці підприємства є загрози в сфері інформаційного забезпечення. Наслідками успішного проведення інформаційних атак можуть стати компрометація або спотворення конфіденційної інформації, нав'язування неправдивої інформації, порушення встановленого регламенту збору, обробки і передачі інформації, відмови і збої в роботі технічних систем, викликані навмисними і ненавмисними

діями як з боку конкурентів, так і з боку інших груп користувачів. До однієї з найбільш важливий завдань в області безпеки підприємства слід віднести створення комплексної системи захисту інформації.

З розвитком інформаційних технологій зростає роль достовірності інформації, що передається по каналам зв'язку. Важливу роль в цій передачі відіграє ідентифікація користувачів на основі цифрового електронного підпису.

ЕЦП, відповідно до стандарту ISO 7498-2, є отримані в результаті криптографічного перетворення блоку даних дані, які дозволяють одержувачу упевнитися в цілісності цього блоку і справжності джерела, а також забезпечує захист від підробки одержувача інформації.

ЕЦП (англ. digital signature) — вид електронного підпису, відповідно до стандарту ISO 7498-2, 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 особу, яка підписувала документ. Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

ЕЦП як спосіб ідентифікації підписувача електронного документу, дозволяє однозначно визначати походження інформації (джерело інформації), що міститься у документі. Завдяки цьому ЕЦП є також надійним засобом розмежування відповідальності за інформаційну діяльність у суспільстві.

Метою магістерської роботи є дослідження програмних засобів формування електронного підпису на основі інформації користувача та обраних алгоритмів кодування.

Об'єкт дослідження – програмні засоби створення електронного цифрового підпису .

Предмет дослідження – алгоритми створення електронного цифрового підпису

Методи дослідження.

Аналіз існуючих традиційних підходів методів створення та захисту ЕЦП та концептуальних складових інформаційної безпеки.

Під час дослідження, що спрямоване на досягнення цієї мети вирішені наступні задачі:

− дослідження традиційних методів створення ЕЦП;

− дослідження існуючих концепцій орієнтованих на безпеку інформації, визначення характеристик та критеріїв підходу для досягнення максимальної ефективності захисту ЕЦП;

− дослідження та вибір оптимального алгоритму створення ЕЦП та підтримки цілісності даних;

− дослідження та вибір програмного комплексу для надійного шифрування ЕЦП та верифікації;

− розроблення складової частини інформаційно-орієнтовного підходу, що забезпечує створення ЕЦП із зашифрованих даних користувача;

− тестування програмних комплексів створення ЕЦП.

Незважаючи на повсякденне використання ЕЦП в перерахованих сферах застосування, на сьогоднішній день склалася досить парадоксальна ситуація - паралельно існують і застосовуються різні державні та комерційні стандарти ЕЦП, при цьому користувач системи ЕЦП зазвичай погано уявляє собі ефективність і якість застосовуваної їм системи, її ступінь криптостійкості і може реально оцінити тільки зручність інтерфейсу програмної реалізації ЕЦП і її швидкодію. Слід зазначити, що багато з застосовуваних систем ЕЦП засновані на обчислювально трудомістких алгоритмах, що викликає відчутні користувачу часові затримки і викликає незадоволеність застосовуваною їм системою.

Незважаючи на велику кількість публікацій з криптографічних протоколів, куди входить і ЕЦП [2,3], залишається невідомий ґрунтовний порівняльний аналіз існуючих і застосовуваних систем ЕЦП. В цьому плані завдання порівняльного дослідження різних алгоритмів і систем ЕЦП по набору критеріїв ефективності є досить актуальним.

Практично всі протоколи ЕЦП використовують криптографічні хеш-функції, що дозволяють шляхом застосування математичного стискає перетворення отримати з документального файлу довільного розміру результат фіксованого довжини - дайджест повідомлення. З метою зменшення обчислювального обсягу ЕЦП і зниження часу на її формування та перевірку алгоритми формування ЕЦП застосовуються до дайджестам, які істотно коротше вихідних повідомлень.

В цьому плані якість системи ЕЦП в значній мірі визначається застосовуваним алгоритмом хеш-функції. Тому порівняльний аналіз систем ЕЦП обов'язково повинен супроводжуватися дослідженням показників ефективності застосовуваних хеш функцій [4].

Критеріями для порівняння систем були обрані зручність програмної реалізації системи ЕЦП та час її роботи, яке як зазначено вище, безпосередньо пов'язано із зручністю користувача.

Під зручністю програмної реалізації розуміється:

– наявність відкритого вихідного коду,

– наявність криптографічних примітивів даної системи ЕЦП у відкритих бібліотеках програмних кодів,

–можливість розпаралелювання алгоритму та наявність інших шляхів прискорення дії алгоритму при його програмній реалізації.

Крипостійкість ЕЦП як і всіх криптографічних протоколів з відкритим ключем не має теоретичного обґрунтування, тому в даному питанні орієнтуємось на концепцію

«практичного кордону рівня безпеки у 80 біт довжини ключа», згідно з якою асиметрична криптосистема з довжиною ключа 80 біт і більше не може бути зламана атакою грубої сили за розумний час [5], а також відповідними стандартами ЕЦП.

Під часом роботи алгоритму електронного підпису будемо розуміти суму витрат часу на операції «генерація ключа підпису», «постановка підпису», «Верифікація підпису». Час роботи залежить від швидкісних якостей криптоалгоритма, що реалізує цифровий підпис і швидкості застосовуваної хеш функції.

В силу істотної нелінійності як алгоритмів криптографічного хешування, так і алгоритмів ЕЦП, не представляється можливим отримати хоча б грубі оцінки обчислювальної складності систем ЕЦП. Тому основним методом їх порівняльного аналізу обрана програмна реалізація різних систем ЕЦП з подальшим їх тестуванням і профілюванням.

Наукова новизна дослідження полягає у подальшому розвитку традиційних методів створення ЕЦП та існуючих програмних концепціях, орієнтованих на підвищення криптозахисту ЕЦП.

Апробація результатів роботи. Основні результати роботи представлені у наступній публікації:

1. Сідєльніков В. В. Програмні засоби створення цифрового електронного підпису / Збірник науково-практичних праць VI молодіжного форуму «ІТ-Ідея 2020» (11 грудня 2020 р.). – Сєвєродонецьк.

Практичне використання. Результати дослідження, запропоновані рішення дозволять створити ЕЦП на основі інформації користувача, а також можуть бути використані у навчальному процесі кафедри комп’ютерних наук та інженерії при вивченні дисципліни «Захист інформації в комп’ютерних системах».

Структура і обсяг роботи.

Магістерська робота складається зі вступу, 5 розділів, висновків, переліку джерел посилань до розділів з 54 найменувань, додатків на 25 сторінках. Загальний обсяг роботи складає 118 сторінок. Магістерська робота містить 19 рисунків та 6 таблиць.

РОЗДІЛ 1
ОГЛЯД ПРОГРАМНИХ ЗАСОБІВ СТВОРЕННЯ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ

1. 1 Нормативно-правова база створення ЕЦП

7 листопада 2018 року набрав чинності Закон України «Про електронні довірчі послуги». З цієї дати втратив чинність Закон України «Про електронний цифровий підпис». Одним з важливих нововведень Закону «Про електронні довірчі послуги» є те, що він вводить поняття «кваліфікованого електронного підпису» (КЕП), яке замінило поняття «електронний цифровий підпис».

Відповідно до Закону України «Про електронні довірчі послуги», кваліфікований електронний підпис - вдосконалений електронний підпис, який створюється з використанням засобу кваліфікованого електронного підпису і базується на кваліфікованому сертифікаті відкритого ключа [2].

Електронний підпис – це електронні дані, які додаються підписантом до інших електронних даних або логічно з ними зв'язуються і використовуються ним як підпис.

Засіб кваліфікованого електронного підпису - апаратно-програмний або апаратний пристрій, або програмне забезпечення, що реалізують криптографічні алгоритми генерації пар ключів і/або створення кваліфікованої електронного підпису та/або перевірки кваліфікованого електронного підпису та/або зберігання особистого ключа кваліфікованого електронного підпису, який відповідає вимогам чинного Закону «Про електронні довірчі послуги».

КЕП накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа. Система цифрового підпису припускає, що кожен користувач мережі має свій особистий ключ (зберігається в таємниці), який використовується для формування підпису, а також відповідний цьому особистому ключу відкритий ключ, відомий решті користувачів мережі, і призначений для перевірки підпису. Цифровий підпис обчислюється на основі особистого ключа відправника інформації та, власне, інформаційних бітів документа (файлу). Спосіб обчислення цифрового підпису гарантує, що знання відкритого ключа не може призвести до підробки підпису.

Для того, щоб мати можливість підписувати електронні документи, подавати електронну звітність або електронні декларації, особа повинна отримати кваліфікований електронний підпис. Видача останньої, відповідно до Закону «Про електронні довірчі послуги», є довірчою послугою, а тому здійснюється кваліфікованими постачальниками

електронних довірчих послуг, перелік яких міститься в довірчому списку, який можна переглянути за посиланням https://czo.gov.ua/trustedlist.

Акредитовані центри сертифікації ключів (АЦСК), утворені відповідно до Закону України «Про електронний цифровий підпис», які надають кваліфіковані електронні довірчі послуги, внесені центральним засвідчувальним органом в Довірчий список як кваліфіковані представники електронних довірчих послуг. Іншими словами, кваліфікований електронний підпис можна отримати і в АЦСК [3].

Наприклад, на сьогодні кваліфіковану електронний підпис можна отримати в:

- АЦСК органів юстиції України;

- АЦСК Інформаційно-довідкового департаменту ДФС;

- інших кваліфікованих представництвах електронних довірчих послуг, перелік яких міститься у «Довірчому списку» (рис. 1.1).

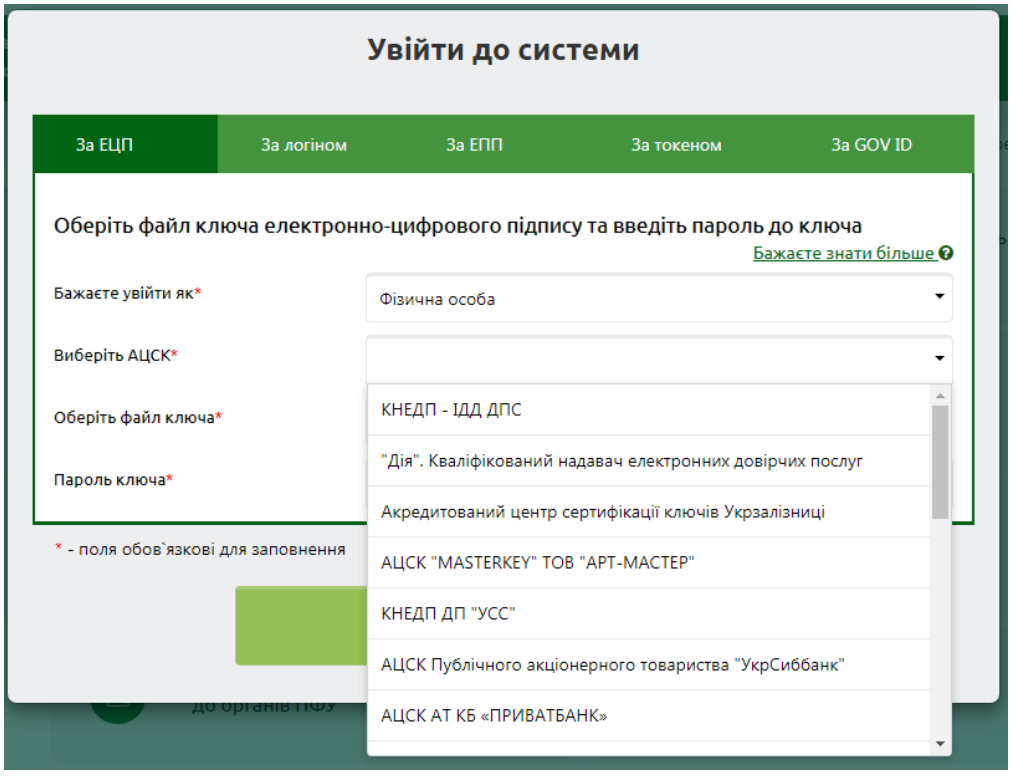


Рисунок 1.1 – Перелік представництв електронних послуг створення ЕЦП

Кваліфікований електронний підпис кожна особа повинна отримувати особисто. Однак, якщо кваліфікована електронний підпис діюча (наприклад, якщо до закінчення терміну його дії залишилося кілька днів), то особа може повторно (дистанційно) сформувати кваліфіковані

сертифікати відкритого ключа з електронним запитом без необхідності персонального присутності в відповідному АЦСК або іншому органі.

Це можна, наприклад, зробити по посиланню. Відповідно до Закону, ідентифікація фізичної особи, яка звернулася за отриманням послуги формування кваліфікованого сертифіката відкритого ключа, здійснюється за умови її особистої присутності за паспортом громадянина України або за іншими документами, які унеможливлюють виникнення будь-яких сумнівів щодо персони, відповідно до законодавства «Про Єдиний державний демографічний реєстр і про документи, що засвідчують особу, що підтверджують громадянство України або спеціальний статус персони ».

Однак, допускається ідентифікація фізичної особи кваліфікованим надає електронні довірчі послуги з ідентифікаційним даним, що містяться в раніше сформованому ним кваліфікованому сертифікаті відкритого ключа, за умови дії цього сертифіката.

Дистанційно сформувати нові сертифікати зможуть лише ті користувачі, які мають:

- діючі сертифікати (наприклад, до закінчення терміну дії сертифікатів залишилося кілька днів)

- незмінні реєстраційні дані (ПІБ, адреса реєстрації місця проживання, код ЄДРПОУ організації і т.п.);

- особистий ключ доступний тільки користувачу і не є розсекречений.

За отримання кваліфікованих сертифікатів відкритих ключів кваліфікованого електронного підпису кваліфіковані представництва електронних довірчих послуг можуть встановлювати плату. Безкоштовно кваліфіковані сертифікати відкритих ключів можна отримати в АЦСК Інформаційно-довідкового департаменту ДФС.

Розмір плати за видачу кваліфікованих сертифікатів відкритих ключів можна дізнатися, зайшовши на офіційний сайт відповідного кваліфікованого постачальника електронних довірчих послуг.

Термін дії кваліфікованих сертифікатів відкритих ключів кваліфікованої електронного підпису становить до двох років з моменту їх формування. Цей термін встановлює відповідний кваліфікований постачальник електронних довірчих послуг (АЦСК).

Відповідно до Закону «Про електронні довірчі послуги», кваліфікована електронний підпис має таку ж юридичну силу, як і власноручний підпис, і має презумпцію відповідності власноручного підпису.

Електронний підпис або печатка не можуть бути визнані недійсними і позбавлені можливості розглядатися як доказ у судових справах виключно на тій підставі, що вони електронний тип або не відповідають вимогам до кваліфікованої електронного підпису або печатки.

Перевірити достовірність кваліфікованої електронного підпису і підтвердити цілісність електронного документа, на який накладено такий підпис, можна на офіційному сайті Центрального засвідчувального органу за посиланням. Для цього потрібно завантажити файл підписаного електронного документа і сертифікат електронного підпису, який був сформований при накладенні такої електронного підпису на електронний документ.

Кваліфіковану електронний підпис можна використовувати в різних сферах електронного обміну інформацією, де необхідно ідентифікувати підписувача, підтвердити цілісність даних і зафіксувати час підписання документа. Кваліфікована електронний підпис використовується, наприклад, при подачі електронних декларацій, підписання договорів, подання електронної звітності, отриманні державних електронних послуг тощо.

Кваліфіковані постачальники електронних довірчих послуг (АЦСК) можуть видавати кваліфіковану електронний підпис на:

- звичайному флеш-носії або оптичному носії CD / DVD (можна принести свою флешку або купити в місці отримання електронного підпису),

- захищеному носії особистих ключів (це може бути, наприклад, захищений носій особистого ключа «Алмаз-1К» виробництва ЗАТ «Інститут інформаційних технологій» або захищений носій особистого ключа «Кристал-1» виробництва ЗАТ «Інститут інформаційних технологій»).

Також кваліфікована електронний підпис може зберігатися на сім-карті особи, в разі отримання послуги Mobile ID.

У разі отримання ЕЦП до вступу Закону України «Про електронні довірчі послуги » в дію існуючий ЕЦП є дійсним і, якщо АЦСК, який його видав не припинив свою діяльність, діє його сертифікат. Відповідно до Закону «Про електронні довірчі послуги», електронний цифровий підпис та посилений сертифікат відкритого ключа, який підтверджує, видані відповідно до вимог Закону України «Про електронний цифровий підпис» до вступу в силу Закону «Про електронні довірчі послуги», використовуються користувачами електронних довірчих послуг, кваліфікованими постачальниками електронних довірчих послуг, які продовжують їх обслуговувати, відповідно як кваліфікована електронний підпис і кваліфікований сертифікат електронного підпису після закінчення терміну дії посиленого сертифікату відкритого ключа, але не пізніше двох років з дня набрання чинності цим Законом «Про електронні довірчі послуги ». Іншими словами, ЕЦП продовжують діяти що два роки після 7 листопада.

Важливо відзначити, що особистий ключ кваліфікованого електронного підпису не можна передавати іншій особі, інакше вони будуть вважатися скомпрометовані, тобто недійсними.

Відповідно до Закону України «Про електронні довірчі послуги» користувачі електронних довірчих послуг зобов'язані:

- забезпечувати конфіденційність і неможливість доступу інших осіб до особистого ключа;

- невідкладно повідомляти постачальнику електронних довірчих послуг про підозру або факт компрометації особистого ключа.

Якщо право підписувати документи передається іншій особі, то така особа може це зробити тільки за допомогою свого власного електронного підпису.

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

1.2 Огляд програмних засобів створення ЕЦП

1.2.1 Trusted eSign ГОСТ компанії «Цифрові технології»

Trusted eSign ГОСТ компанії «Цифрові технології» [4] призначений для цифрового підпису будь-яких файлів під операційними системами Linux і macOS, використовуючи відомі крипто алгоритми (рис. 1.2).

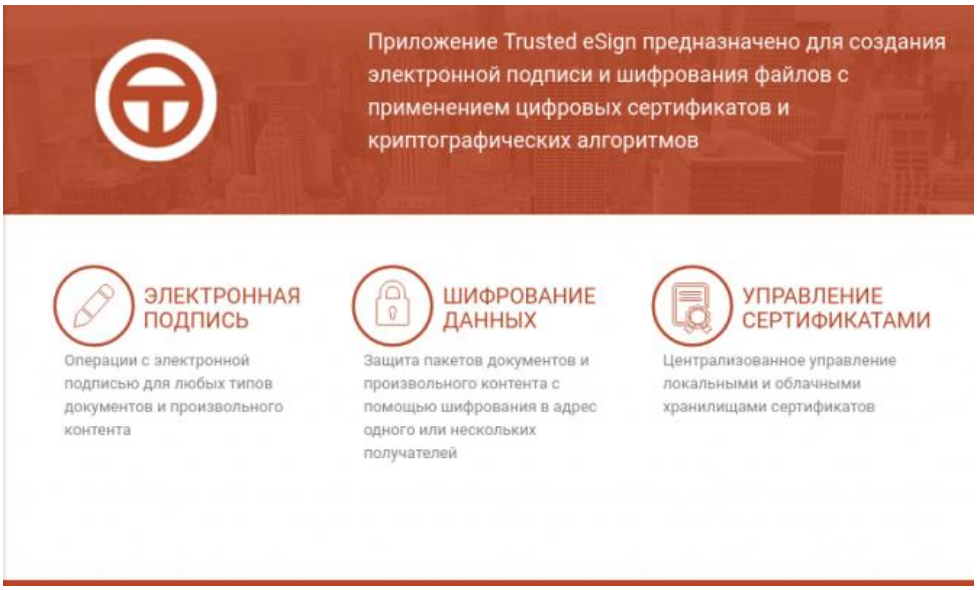


Рисунок 1.2 – Загальне вікно програми Trusted eSign ГОСТ компанії «Цифрові технології»

Trusted eSign ГОСТ має простий і зрозумілий інтерфейс, підтримує найбільш поширені ключові носії. Продукт внесений до реєстру програмного забезпечення.

У додатку до програми наголошено, що підтримані раніше стандарти для електронного підпису і хешировання ГОСТ Р 34.10-2012 і ГОСТ Р 34.11-2012 з 01 січня 2019 року припинили свою дію разом зі стандарти ГОСТ Р 34.10-2001 і ГОСТ Р 34.11-94.

Trusted eSign ГОСТ – це можливість підписати будь-які файли за допомогою електронного підпису в простому user-friendly інтерфейсі. Крім підпису, в програмі також можна перевірити достовірність електронного підпису, зашифрувати важливі для вас дані або розшифрувати їх. Додаток створено на сучасному движку Electron, для виклику криптографічних операцій застосовується бібліотека OpenSSL. Всі необхідні компоненти поставляються в складі установчого пакета.

Ціна програмного забезпечення – 30 $. за одне робоче місце. Платформа Linux, Windows. Ліцензування – одноразово.

Схематично процес створення електронного цифрового підпису наведено на рисунку 1.3.

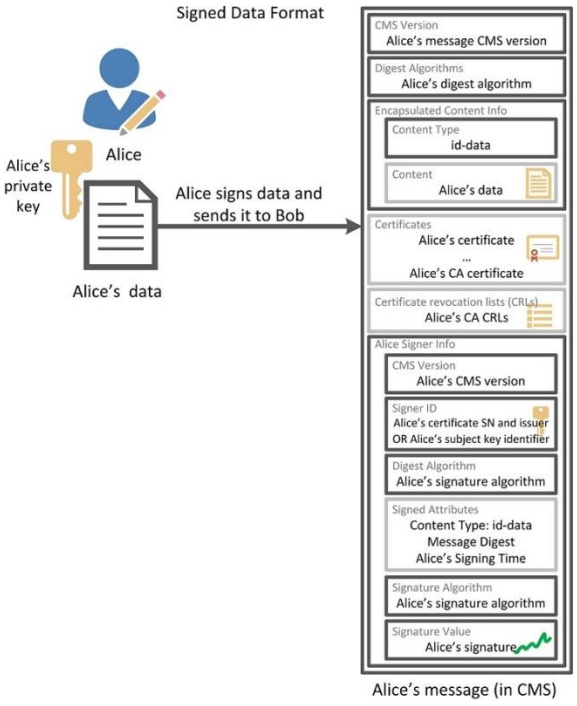


Рисунок 1.3 – Процес створення електронного цифрового підпису

Ключові функції – створення та перевірка електронного підпису стандарту CMS (Cryptographic Message Syntax) [5].

Цей стандарт передбачає, що дані можуть бути підписані декількома сторонами, причому тип даних не регламентується. Збереження результатів підпису в кодуваннях

Base64, DER. Шифрування даних на адресу сертифікатів одержувачів і розшифрування. Архівування файлів перед шифруванням і видалення вихідних файлів після шифрування. Установка ключів і сертифікатів за стандартами PKCS # 8 і x.509 v3 [6].

1.2. 2 Litoria Crypto Platform

Криптографічна платформа Litoria Crypto Platform увібрала в себе досягнення фахівців компанії в напрямку інфраструктури з відкритими ключами більш ніж за 10 років існування на ринку інформаційної безпеки (рис. 1.4) [7].

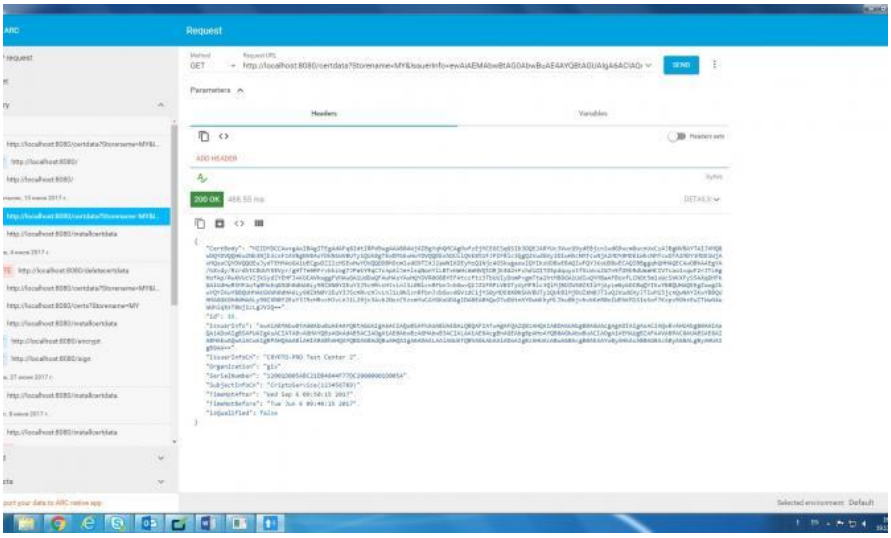


Рисунок 1.4 – Загальний вигляд криптографічної платформи Litoria Crypto Platform

Інфраструктура відкритих ключів (англ. *Public key infrastructure*, PKI) — інтегрований комплекс методів та засобів (набір служб), призначених забезпечити впровадження та експлуатацію криптографічних систем із відкритими ключами [8].

Вона відповідає вимогам в частині реалізації технологій ЕП і шифрування і при цьому враховує кращі світові практики (успішно проходить всі тести PKI, розроблені Національним інститутом стандартів і технологій США - NIST). Рік від року число державних відомств і комерційних структур, які здійснюють перехід до електронного юридично значущого і конфіденційної взаємодії, неухильно зростає.

Ключові функції створення, додавання, завірення і перевірка ЕП різного типу (проста, вдосконалена ЕП (УЕП), відокремлена); створення і додавання ЕП документа без надання даних (підпис хеш-значення документа); шифрування, розшифрування, гарантоване видалення файлів; управління СОС: створення, імпорт, експорт, видалення, детальний

перегляд; застосування для роботи з ЕП різних криптопровайдерів, як програмних (Base CSP, КріптоПро CSP, ViPNet CSP, ВАЛІДАТА CSP, Avest CSP і ін.), так і апаратних (JaCarta, eToken ГОСТ, РУТОКЕН ЕЦП); реалізація функцій служби актуальних статусів сертифікатів в Litoria Crypto Platform згідно RFC2560 Online Certificate Status Protocol (OCSP); реалізація функцій служби штампів часу згідно RFC3161 Time-Stamp Protocol (TSP); формування і перевірка ЕП електронних повідомлень відповідно до рекомендацій RFC3029 Data Validation and Certification Server Protocol (DVCS); інтеграція сервісів пролонгації (для реалізації сервісів пролонгації УЕП підтримується стандарт CAdES-A); інтеграція Litoria Crypto Platform в системи електронного документообігу, включаючи SharePoint, Citrix, веб-сервер MS IIS (Internet Information Servies); надання інтерфейсів (Com, Java, C #, SilverLight, ASP.net) для вбудовування в різні середовища і системи.

Платформа Linux, Windows, macOS, iOS.

1.2.3 КриптоПро DSS

Програмно-апаратний комплекс КриптоПро DSS призначений для централізованого, захищеного зберігання закритих ключів користувачів, а також для віддаленого виконання операцій зі створення електронного підпису (ЕП) [9].

Створення і зберігання ключів електронного підпису користувачів здійснюється з використанням спеціального захищеного модуля КриптоПро HSM. Кожен користувач отримує доступ до своїх ключам після проходження процедури надійної багатофакторної аутентифікації в КриптоПро DSS.

Додатково кожен ключовий контейнер захищається індивідуальним ПІН-кодом, який знає і може змінити тільки власник ключа електронного підпису.

КриптоПро DSS надає користувачам інтерактивний веб-інтерфейс для управління криптографічними ключами і створення ЕП під документом, який користувач завантажує на КриптоПро DSS.

Таким чином, для роботи з КриптоПро DSS користувачеві необхідний тільки веб-браузер, ніяких ЗКЗІ або засобів електронного підпису встановлювати не потрібно. Завдяки цьому, використовувати функції КриптоПро DSS можна з будь-якого пристрою з будь-апаратною платформою з будь-якою операційною системою, де є веб-браузер (рис. 1.5).

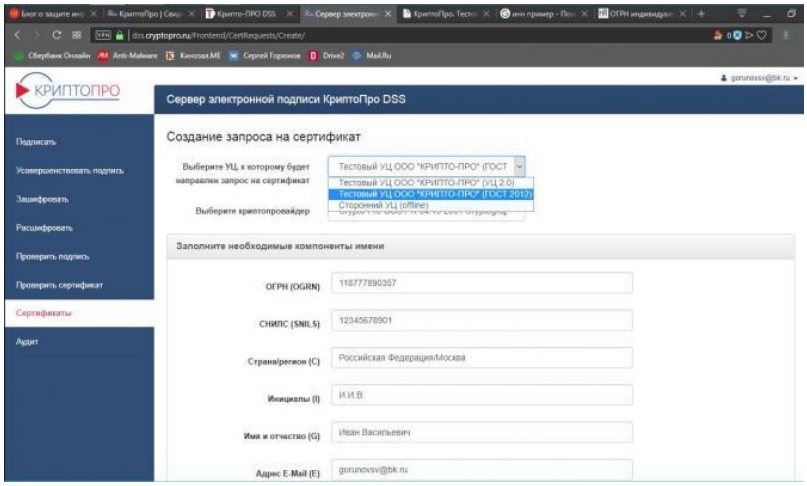


Рисунок 1.5 – Загальний вигляд програми КриптоПро DSS

Ключові функції створення користувача; блокування та видалення користувача; генерація ключа електронного підпису користувача; формування та передача в засвідчує центр запиту на створення сертифіката ключа перевірки електронного підпису; установку отриманого сертифіката користувачеві; налаштування параметрів аутентифікації користувачів; аудит і формування аналітичної звітності по виконуваних користувачами операціями; скидання пароля в разі його втрати користувачем.

Платформа – платформа SaaS (Програмне забезпечення як послуга), Android, Windows, iOS [10].

1.2.4 КриптоАРМ

КриптоАРМ призначена для захисту корпоративної та особистої інформації, переданої по мережі Інтернет, електронною поштою і на знімних носіях (дисках, флеш-картах).

На цей час надається можливість безкоштовно ознайомитися з програмою в повному обсязі без обмежень її функціональності. Ознайомчий період активується автоматично тільки одного разу при першій установці програми на робочому місці. "КриптоАРМ SDK" - бібліотека криптографічних функцій (створена у відповідності зі стандартами компонентної моделі компанії Microsoft). У комплект поставки входить документація для розробників, де описуються способи інтеграції "КриптоАРМ" з зовнішніми програмами на рівні програмного коду, наведені призначення, характеристики, функції програми, а також приклади використання основних операцій.

У КриптоАРМ є 14-денний термін для ознайомлення з програмою (рис. 1.6) [11].

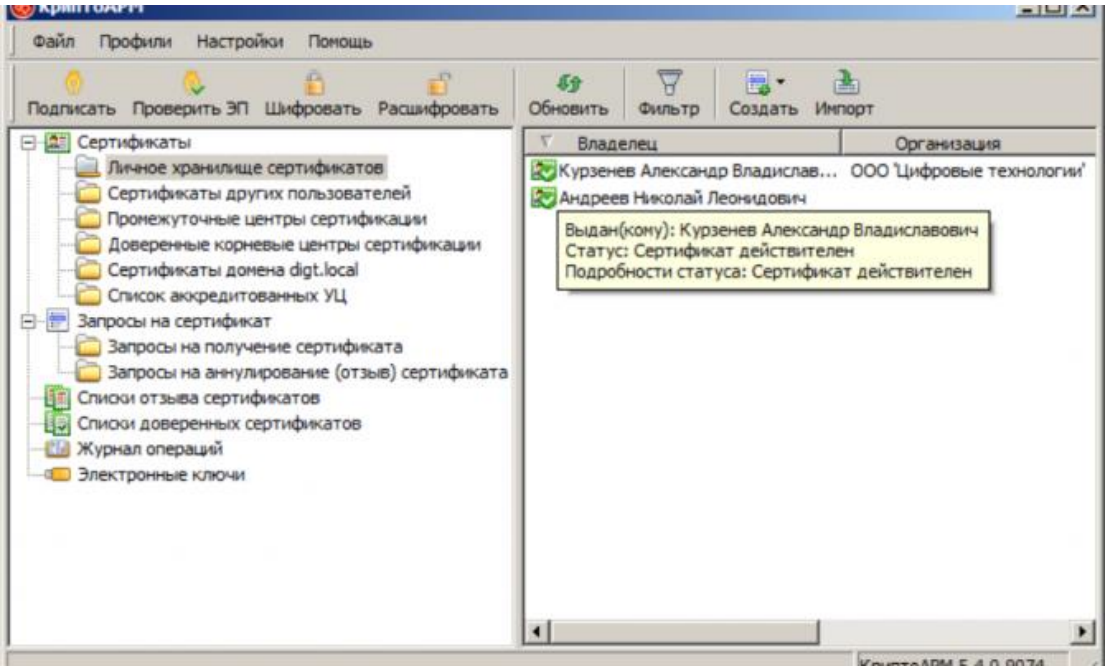


Рисунок 1.6. – Загальний вигляд програми КриптоАРМ

Ключові функції – Надійний захист даних від стороннього доступу, гарантія цілісності даних при відправці по незахищених каналах зв'язку, забезпечення достовірності та авторства електронних документів, узгодження електронні документи з колегами.

Вартість програми – 13 $ на 1 рік за 1 робоче місце.

Платформа – Windows.

1.3 Недоліки існуючих схем формування ЕЦП

– повільна робота алгоритмів формування і перевірки підпису;
– обмеження на довжину повідомлення, яке підписується.

Одне з рішень проблеми, пов'язаної з обмеженнями на довжину, – розбиття повідомлення на фрагменти і підпис кожного фрагмента.

Однак таке рішення часто неприйнятно для використання на практиці, так як результатом буде збільшення обсягу повідомлення і часу виконання процедур створення і перевірки ЕЦП.

1.4 Застосування функцій хешування для ЕЦП

Для зменшення часу, необхідного для генерації і перевірки підпису, а також для скорочення її довжини застосовується механізм хеш-функції [12].

Відображення підписується повідомленням, отриманим в результаті застосування хеш-функції h та має мати невелику фіксовану довжину, багато меншу довжини самого повідомлення.

Тоді підписане повідомлення m матиме вигляд $(m, S(h(m))$, де S - функція вироблення підпису, причому функція h повинна бути односторонньою, тобто не повинно існувати алгоритму поліноміальної складності для обчислення m за відомим $h(m)$.

1.5 Надійність практичних реалізацій схем створення та перевірки ЕЦП

Фахівцями достатньо вивчені питання ненадійності практичних реалізацій алгоритмів формування і перевірки ЕЦП.

Аналіз вразливостей існуючих схем ЕЦП дозволяє фахівцям стверджувати, що «число вразливих точок ЕП, що базується на шифруванні з відкритим ключем, настільки велике, що доцільність використання подібного методу викликає великі сумніви».

Основною причиною вразливості ЕЦП є вразливість алгоритму шифрування з відкритим ключем, що лежить в основі технології електронного підпису.

Друга причина - передача відкритого ключа в одному конверті з електронним підписом (у структуру ЕЦП, згідно з міжнародним стандартом CCITT X.509 [13], входить не тільки відкритий ключ відправника, а й його ім'я, серійний номер ЕЦП, назва і власна ЕЦП уповноваженої організації, що видала набір секретного і відкритого ключа).

В даний час реалізовані і опубліковані схеми механізмів злому ЕЦП, засновані на генерації нової пари (відкритий, секретний) ключів і включення нового відкритого ключа в конверт ЕЦП.

Надійність системи ЕЦП складається з надійності окремих елементів, до яких крім алгоритмів вироблення і перевірки підпису відносяться механізм генерації і розподілу ключів і ряд інших елементів. На надійність системи ЕЦП важливий вплив надає розподіл ключів між абонентами, які беруть участь в обміні повідомленнями.

На практиці такий розподіл здійснюється двома способами:

– створенням центру генерації і розподілу ключів;

– прямим обміном ключами між абонентами.

У першому випадку компрометація центру призводить до компрометації всієї інформації, що передається. У другому випадку - необхідно забезпечити справжність кожного абонента.

Помилки реалізації систем ЕЦП також істотно впливають на зниження рівня надійності схем.

Поширеними помилками є:

1) періодичне повторення одних і тих же значень, одержуваних поширеними алгоритмами генерації випадкових чисел;

Справжні випадкові числа не повинні бути передбачувані.

Генератори випадкових чисел видають псевдовипадкові числа (при кожному виконанні алгоритму генерації випадкових чисел отримаємо той же самий список випадкових чисел).

Тому часто на практиці використовують спеціальну апаратуру для генерації справжніх випадкових чисел.

Наприклад, ОС Unix (Linux) мають спеціальний код в складі драйверів пристроїв нижнього рівня, який безперервно збирає випадкові сигнали клавіатурних вводів та інших апаратних пристроїв і надає їх додатків в якості джерела випадкових чисел.

2) наявність колізій (можливість генерації однаковою хеш-функції для різних повідомлень);

3) розробка власних алгоритмів, що володіють властивостями якісних криптографічних алгоритмів.

1.6 Висновки до розділу 1

У першому розділі магістерської роботи проведено огляд та порівняльний аналіз програмних засобів створення електронного цифрового підпису.

Розглянута нормативно-правова база створення ЕЦП.

Наголошено, що відповідно до Закону України «Про електронні довірчі послуги», кваліфікований електронний підпис – вдосконалений електронний підпис, який створюється з використанням засобу кваліфікованого електронного підпису і базується на кваліфікованому сертифікаті відкритого ключа.

Схеми формування ЕЦП, що базуються на шифруванні з відкритим ключем, принципово уразливі.

Ефективність використовуваних на практиці схем формування ЕЦП, заснованих на криптографії з відкритим ключем, з точки зору швидкодії, є досить низькою.

Сучасні практичні реалізації схем ЕЦП є вразливими.

З огляду на бурхливий розвиток обчислювальних потужностей сучасних комп'ютерних систем і математичних методів криптоаналізу, практична схема цифрового підпису повинна гарантувати достатній рівень захисту на роки вперед.

При використанні ЕЦП об'єктом захисту поряд з самим об'єктом є і його ЕЦП.

Серед недоліків існуючих схем формування ЕЦП відмічається:

– повільна робота алгоритмів формування і перевірки підпису;

– обмеження на довжину повідомлення, яке підписується.

Відомі програмні рішення пов'язані з обмеженнями на довжину, розбиття повідомлення на фрагменти і підпис кожного фрагмента.

Однак таке рішення часто неприйнятно для використання на практиці, так як результатом буде збільшення обсягу повідомлення і часу виконання процедур створення і перевірки ЕЦП.

В результаті дослідження намічені подальші шлях удосконалення алгоритмів ЕЦП, направлені на зменшення кількості операцій кодування та збільшення криптографічної стійкості ЕЦП.

1.7 Перелік посилань до вступу та розділу 1

1. Протокол Діффі-Геллмана. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Протокол_Діффі_—_Геллмана (дата звернення 24.10.2020).

2. Закон України «Про електронні довірчі послуги». [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2155-19#Text (дата звернення 24.10.2020).

3. Інформація для кваліфікованих надавачів електронних довірчих послуг. [Електронний ресурс]. – Режим доступу: http://www.ukrstat.gov.ua/elektr_zvit/inf_akcen/centr.htm (дата звернення 24.10.2020).

4. Средства электронной подписи. Trusted eSign ГОСТ. [Електронний ресурс]. – Режим доступу: https://www.anti-malware.ru/products/trusted-esign-gost (дата звернення 24.10.2020).

5. Форматы электронной подписи. Для чего нужен CMS. [Електронний ресурс]. – Режим доступу: https://habr.com/ru/company/aktiv-company/blog/191866/ (дата звернення 24.10.2020).

6. Формат сертификатов. Формальное описание структур даннях. [Електронний ресурс]. – Режим доступу: https://parallel.uran.ru/book/export/html/526 (дата звернення 24.10.2020).

7. Litoria Crypto Platform. [Електронний ресурс]. – Режим доступу: https://www.gaz-is.ru/produkty/dokumentooborot/litoria-crypto-platform.html (дата звернення 24.10.2020).

8. Інфраструктура відкритих ключів, що побудована на сертифікатах. [Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5367447/ (дата звернення 24.10.2020).

9. Програмно-апаратний комплекс КриптоПро DSS. [Електронний ресурс]. – Режим доступу: https://www.anti-malware.ru/products/cryptopro-dss (дата звернення 24.10.2020).

10. Что такое SaaS и как это работает. [Електронний ресурс]. – Режим доступу: https://blog.ringostat.com/ru/chto-takoe-saas-i-kak-eto-rabotaet/ (дата звернення 24.10.2020).

11. КриптоАРМ. [Електронний ресурс]. – Режим доступу: https://www.anti-malware.ru/products/cryptoARM (дата звернення 24.10.2020).

12. Механізм хеш-функції. [Електронний ресурс]. – Режим доступу: http://mf.grsu.by/UchProc/livak/b_protect/zok_7.htm (дата звернення 24.10.2020).

13. Стандарт CCITT X.509. [Електронний ресурс]. – Режим доступу: https://proverkassl.com/docs_x.509.html (дата звернення 24.10.2020).

РОЗДІЛ 2
АЛГОРИТМИ ТА МОДЕЛІ КРИПТОСИСТЕМИ З ВІДКРИТИМИ КЛЮЧАМИ

2.1 Односторонні функції з секретом і асиметричні системи

Найслабшою ланкою при реалізації симетричних криптосистем в системах захищеного електронного документообігу, електронних банківських платежів і, особливо, електронної торгівлі є питання розподілу ключів [1].

Для забезпечення обміну конфіденційною інформацією між двома абонентами телекомунікаційної мережі повинен бути згенерований ключ (можливо, одним з абонентів), а потім по деякому захищеному каналу переданий іншим користувачам (іншому абоненту).

Як засіб створення захищеного каналу знову може бути використана деяка криптосистема.

Найбільшу гостроту питання розподілу і доставки ключів набуває у разі неможливості наперед описати склад інформаційно-телекомунікаційної мережі.

Для вирішення цієї проблеми на основі результатів, одержаних класичною і сучасною алгеброю, були запропоновані системи з відкритим ключем (СВК).

В СВК для зашифрування не використовуються секретні ключі – вони необхідні тільки при розшифрування (рис. 2.1) [2].

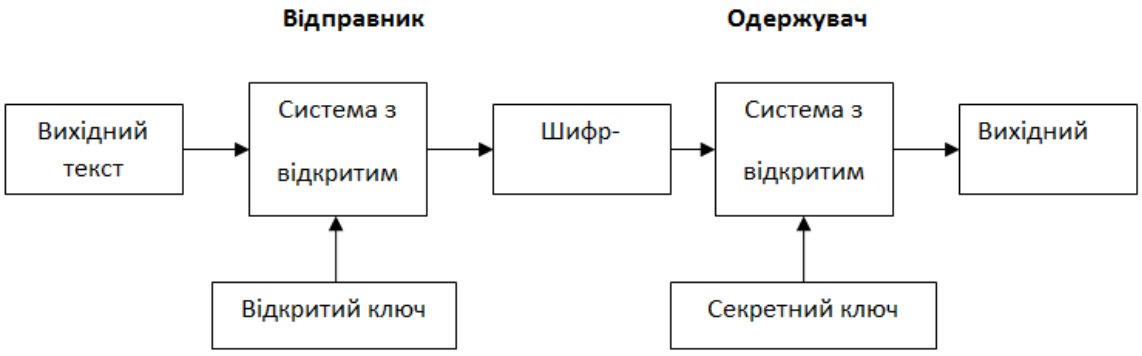


Рисунок 2.1 - Криптосистема з відкритим ключем

Головним поняттям СВК є однонаправлена функція з секретом (one-way trapdoor function).

Однонаправлену функцію з секретом $f_t(x)$: $D \rightarrow R$ легко обчислити для всіх $x \in D$, але дуже важко обернути майже для всіх значень із R.

Однак, якщо використовується деяка секретна інформація t, то для всіх значень y, що належать R легко обчислити величину $x \in D$, що задовольняє умову $y = f_t(x)$ [3].

Криптосистеми з відкритим ключем, що використовують однонаправлені функції з секретом, ще називаються асиметричними (asymmetric cryptosystems).

В 1975 році Діффі і Хеллман в роботі, присвяченій криптографії з відкритим ключем, запропонували декілька варіантів побудови однонаправлених функцій з секретом.

Проте ці функції не були точно асиметричними, тому незабаром Діффі і Хеллман запропонували більш вдалий варіант: показникову функцію за простим модулем [4].

Ця функція була використана в широко розповсюдженому криптографічному протоколі – протоколі обміну ключами Діффі-Хеллмана (Diffie-Hellman key exchange protocol).

Раніше, в 1974 році, Меркл (Merkle) винайшов механізм узгодження криптографічного ключа шляхом очевидних асиметричних обчислень, що одержали назву головоломка Меркла [5].

Асиметричність головоломки Меркла полягає в тому, що її обчислювальна складність для законних учасників протокола узгодження ключа і для перехоплювачів зовсім різна: легальні участники легко виконують обчислення, а нелегальні — ні. Головоломка Меркла – це перша ефективна реалізація однонаправленої функції з секретом.

Не дивлячись на те, що головоломка Меркла не підходить для застосування в сучасних криптографічних додатках, її вплив на криптосистеми з відкритим ключем неможливо переоцінити.

Останнім часом стало відомо, що першу криптосистему з відкритим ключем розробив британський математик Кокс (Cocks) ще в 1973 році.

Алгоритм Кокса, що одержав назву алгоритму з несекретним ключем шифрування, використовував складність розкладання цілого числа на прості множники і, по суті, збігався з криптосистемою RSA [6].

Спочатку алгоритм Кокса був засекречений і лише в грудні 1997 року Група електронного захисту засобів зв'язку (Communications Services Electronic Security Group – CESG) його розсекретила.

Не дивлячись на те, що спочатку криптосистеми з відкритим ключем були надбанням вузького кола осіб, саме завдяки відкритим дослідженням вони знайшли два найважливіші

застосування для створення алгоритмів цифрового підпису і механізмів обміну секретними ключами через відкриті канали зв'язку.

В наш час алгоритми шифрування з відкритим ключем отримали широке розповсюдження в інформаційних системах.

Так, алгоритм RSA фактично став стандартом для відкритих систем і рекомендований міжнародною організацією зі стандартизації ISO, відповідні додатки лежать в основі електронної комерції, здійснюваної через Internet.

Суть їх полягає в тому, що кожним абонентом мережі генеруються два ключі, зв'язані між собою за певним математичним законом. Один ключ оголошується відкритим, а інший закритим (секретним).

Відкритий ключ опубліковується і доступний будь-кому, хто бажає послати повідомлення власнику секретного ключа.

Секретний ключ зберігається в таємниці.

Вихідний текст шифрується відкритим ключем Одержувача і передається йому. Зашифрований текст в принципі не може бути розшифрований тим же відкритим ключем. Розшифрувати повідомлення можна тільки з використанням секретного ключа, який відомий лише самому адресату.

Таким чином, одностороння функція з секретом, що використовується в асиметричній системі, є взаємооднозначною, але разом з тим має властивості необерненості.

Необерненість, звичайно, розуміється не в загальноприйнятому значенні, а як практична неможливість обчислити обернену функцію, використовуючи сучасні обчислювальні засоби за досяжний інтервал часу.

Тому, щоб гарантувати надійний захист інформації, до СВК висуваються дві важливі і очевидні вимоги.

1. Перетворення вихідного тексту повинно бути обчислювано нереалізованим (необернениим) і уникати його відновлення на основі відкритого ключа.

2. Визначення закритого ключа на основі відкритого також повинно бути неможливим на сучасному технологічному рівні. При цьому бажана точна нижня оцінка складності (кількості операцій) розкриття шифру.

В основі сучасних криптосистем з відкритим ключем обчислювально необернені перетворення частіше за все будуються на основі таких алгоритмічних проблем:

– розкладання великих чисел на прості множники;
– обчислення логарифма в скінченному полі;
– знаходження кратності точки (дискретний логарифм) на еліптичній кривій;

– обчислення коренів алгебраїчних рівнянь над кільцями і полями.

Відзначимо області вживання СВК:

1. Засоби шифрування даних, що передаються і зберігаються.

2. Засоби аутентифікації користувачів і перевірки цілісності даних, формування електронного цифрового підпису.

3. Механізми розподілу ключів.

Алгоритми СВК працюють повільніше, ніж алгоритми симетричних шифросистем. Тому на практиці переважним є комбіноване використовування СВК і симетричних систем.

При цьому СВК використовується для створення механізмів розподілу ключів, об'єм яких незначний, а за допомогою симетричних алгоритмів здійснюється шифрування великих інформаційних потоків.

Найбільше розповсюдження отримали системи з відкритим ключем на основі алгоритму RSA, криптосистема Ель-Гамаля [7] і криптосистеми на основі еліптичних кривих [8].

2.2 Криптосистема RSA

Криптосистема RSA, розроблена в 1977 році, одержала свою назву на честь її авторів: Рона Рівеста, Аді Шаміра і Леонарда Ейдельмана.

Розробники скористалися тим фактом, що знаходження великих простих чисел в обчислювальному відношенні здійснюється достатньо просто, але не відомий алгоритм, що виконує за поліноміальний час розкладання на прості множники великих чисел.

Доведено (теорема Рабіна), що розкриття шифру RSA еквівалентне знаходженню такого розкладу.

Тому для будь-якої довжини ключа можна дати (сучасну практичну) нижню оцінку числа операцій для розкриття шифру, а з урахуванням продуктивності сучасних комп'ютерів оцінити і необхідний на це час.

Можливість реально оцінити захищеність алгоритму RSA стала однією з причин популярності цієї СВК на фоні десятків інших схем.

Тому алгоритм RSA використовується в банківських комп'ютерних мережах, особливо для роботи з віддаленими клієнтами (обслуговування кредитних карток).

В наш час алгоритм RSA використовується в багатьох стандартах, серед яких SSL, S-HTTP, S-MIME, S/WAN, STT і PCT.

Крім того, алгоритм RSA реалізується як у вигляді самостійних криптографічних продуктів (PGP), так і як вбудовані засоби в деяких додатках (Інтернет- браузери від Microsoft і Netscape).

Нагадаємо ряд положень елементарної теорії чисел, що лежать в основі цього алгоритму.

Найбільшим спільним дільником двох цілих чисел a і b називається найбільше ціле число, яке ділить як a, так і b.

Позначення: $(a, b)=$ або НСД $(a, b)=$. Числа a і b називаються взаємно простими, якщо $(a, b)=1$.

Важливим фактом є те, що НСД $(a, b)=$ можна виразити за допомогою рішень діофантового рівняння.

Нехай $a>b$ і $d=(a,b)$. Тоді існують цілі числа x, y, що є розв'язками рівняння $xa+yb=d$.

Розв'язки x, y, d рівняння $xa+yb=d$., за умов $a>b$ і $d=(a,b)$ можна знайти з допомогою так званого розширеного алгоритму Евкліда.

Очевидно, достатньо розв'язати рівняння при додатних a і b.

Отже, для взаємно простих чисел m і b, $m>b$ можна знайти числа x, y такі, що $xm+yb=d=1$.

Приведемо останню рівність за модулем m, отримаємо $yb\equiv 1 \bmod m$.

Побудоване число y називається числом, оберненим до b за модулем m і позначається через $y\equiv b^{-1} \bmod m$.

Таке число в діапазоні $(1,\ldots,m-1)$ є єдиним.

Відмітимо, що з цієї причини множина H всіх найменших невід'ємних лишків, взаємно простих з m, при множенні на будь-який свій елемент h піддається перестановці: $h\cdot H=H$.

Розглянемо степені числа a за модулем m, де a і m взаємно прості.

Нехай $m=1$. Степені 1, 2, …10 числа 2 такі: 2, 4, 8, 5, 10, 9, 7, 3, 6, 1.

Аналогічно, степені числа 3 дорівнюють 3, 9, 5, 4, 1, 3, 9, 5, 4. В кожному випадку є періодичність/

Найменша довжина періоду для числа a за модулем m називається порядком (показником, періодом) числа a за модулем m.

Порядок числа a за модулем m позначається $ord_m a$.

Порядки чисел за модулем m різні. Існують числа, що є порядком одночасно для всіх чисел, взаємно простих з m.

Одне з них рівне значенню так званої функції Ейлера $\varphi(m)$, яка визначається як кількість чисел в послідовності $1, \ldots, m$, взаємно простих з m.

З означення слідує, що $\varphi(p)=p-1$ для простого числа p і, крім того, $\varphi(1)=1$.

Функція Ейлера є мультиплікативною: якщо $(a,b)=1$, то $\varphi(ab)=\varphi(a)\,\varphi(b)$.

Теорема Ейлера. Якщо $(a,m)=1$, то $a^{\varphi(m)} \equiv 1 \bmod m$.

Доведення. При $m=2$ теорема справедлива.

Нехай $m>2$ і $a_1,\ldots a_k$ – всі лишки за модулем m, взаємно прості з модулем (за означенням, $k=\varphi(m)$).

Нехай $a>1$ – один з таких лишків. Тоді множини $\{a_1,\ldots a_k\}$ і $\{aa_1 \bmod m,\ldots, aa_k \bmod m\}$ збігаються.

Тому $a^{\varphi(m)} a_1 \cdots a_k = a_1 \cdots a_k \equiv b \bmod m$. Помноживши обидві частини порівняння на $b^{-1} \bmod m$, одержимо $a^{\varphi(m)} \equiv 1 \bmod m$

З теореми Ейлера слідує мала теорема Ферма: $a^{p-1} \equiv 1 \bmod p$, де p – просте, $a \not\equiv 0 \bmod p$.

Випадок $a \equiv 0 \bmod p$ можна врахувати у виразі $a^p \equiv a \bmod p$.

Існує загальна формула для $\varphi(m)$ при відомому канонічному розкладі m на степені простих чисел.

Нехай $m=p_1^{a},p_2^{b},\ldots p_s^{t}$, тоді $\varphi(m)=p_1^{a-1},p_2^{b1-1},\ldots p_s^{t-1},(p_1-1)\ldots(p^s-1)$.

Таким чином, якщо $n=pq$, де p,q нерівні прості числа, то $\varphi(n)=(p-1)(q-1)$.

Для модулів n вказаного вигляду можна показати, що якщо e число, взаємно просте з $\varphi(n)$, то відображення $E_{e,n}$: $x \to x^e \bmod n$ є взаємно однозначним на множині лишків за модулем n.

При цьому, $ed \equiv 1 \bmod \varphi(n)$, а оберненим відображенням є $E_{d,n}$: $(x^b)^d \to x \bmod n$.

В криптосистемі RSA зашифрування блоку повідомлення m проводиться за формулою $c=E_{b,n}(m)$, а для розшифрування застосовується операція $m=E_{d,n}(c)$.

Таким чином, відкритим і секретним ключем криптосистеми є, відповідно, (e,n) і d.

Побудова ключа d при відомих e, d, p, q легко здійсненна. За наявності відповідних параметрів функції $E_{e,n}$ і $E_{d,n}$ також легко обчислюються.

Якщо відомі e і n, але p і q невідомі, то $E_{e,n}$ являє собою односторонню функцією з секретом.

Побудова $E_{d,n}$ за заданими e і n рівносильна розкладанню числа n на співмножники. У разі, коли p і q – достатньо великі прості числа, то розкладання n практично неможливе. Це і є причиною стійкості криптосистеми RSA.

Розглянемо принципи організації інформаційного обміну з використанням системи RSA. Спочатку абонент i вибирає пару різних простих чисел p_i і q_i. Потім він обчислює $n_i=p_iq_i$ і, вибирає випадковий лишок e_i, взаємно простий з $\varphi(n_i)$, і знаходить $d_i \equiv e_i^{-1} mod \varphi(n_i)$

На загальнодоступному сервері (сайті) розміщується довідкова таблиця, яка містить відкриті ключі абонентів (e_i,n_i).

Для передачі криптограми від абонента i до абонента j абонент i розбиває відкритий текст на блоки і послідовно зашифровує їх за допомогою перетворення $E_{bj,nj}$. Абонент j проводить розшифрування поблочно, застосовуючи відображення $E_{dj,nj}$.

Очевидно, для того, щоб знайти d_i, достатньо знання співмножників p_i і q_i Для сучасних технологічних можливостей час виконання найкращих з відомих алгоритмів розкладання для значень $n=2^{1024}$ дуже великий.

Розглянемо навчальний приклад, що ілюструє застосування алгоритму RSA. Зашифруємо повідомлення m=3,2,1, що складається з трьох блоків.

1. Виберемо прості числа: p=3, q=11.

2. Обчислимо $n=pq=33$ і $\varphi(n)=(p-1)(q-1)=20$.

3. Виберемо випадкове значення $e=7$, $e,n=1$.

5. Обчислимо секретний ключ $d=3$, $ed= 1 \ mod(n)$

Відкритий ключ – (7,33).

Зашифруємо повідомлення: $m=3,2,1$.

RSA(3)= 3^7=2187=9mod33;

RSA(2)= 2^7=128=29mod33;

RSA(1)= 1^7=1mod33.

Для розшифрування піднесемо кожний блок до степеня $d=3$ за модулем 33:

9^3=729≡3mod33, 1^3≡1mod33, 29^3=24389≡2mod22

Секретний ключ для навчальної системи легко знайти перебором.

На практиці це неможливо, оскільки реальний розмір модуля (довжина бітового подання) $size(n)$ знаходиться в діапазоні від 512 до 4096 бітів.

Основні зусилля в ході практичної реалізації RSA припадають на генерацію випадкових великих простих чисел.

Є очевидний шлях рішення задачі: випадково вибрати велике непарне число n і перевірити його подільність на множники в діапазоні 3,…,√n.

У разі невдачі вибираємо число $n+2$ і так далі.

Проте при великих n такий підхід нездійсненний.

Помітимо, що з теорії чисел відомо, що ймовірність того, що навмання вибране число порядку n буде простим, оцінюється як $1/\ln(n)$.

В принципі за p і q можна використовувати «майже» прості числа, тобто такі числа, для яких ймовірність того, що вони прості, наближається до 1.

Але у випадку, якщо використано складене число, а не просте, криптостійкість RSA падає.

Є непогані алгоритми, які дозволяють генерувати «майже» прості числа з досить малою ймовірністю помилки.

Інша проблема – якої довжини слід використовувати ключі?

Корисно навести довжини параметрів RSA в бітах, встановлені і рекомендовані французькими фахівцям для практичних додатків.

1. Співмножники RSA-модуля $n=pq$ повинні вибиратися випадково і бути однакової довжини.

2. Довжина секретного ключа d повинна бути порівнянна з розміром модуля n.

3. Довжина відкритого ключа e повинна бути строго більше 16 бітів.

4. До 2010 року дозволялося використовувати значення модуля довжиною не менше 1536 бітів, проте рекомендується – не менше 2048 бітів.

5. З 2010 року по 2020 рік дозволяється використовувати значення модуля довжиною не менше 2048 бітів.

6. Після 2020 року передбачається використовувати значення модуля довжиною не менше 4096 бітів.

Наступний важливий аспект реалізації RSA – обчислювальний. Адже доводиться використовувати апарат арифметики великих чисел.

Слід враховувати, що, в порівнянні з тим же алгоритмом DES, для RSA потрібен в тисячі і десятки тисяч раз більший час.

2.3 Криптосистема Ель-Гамаля

На відміну від RSA асиметрична криптосистема Ель-Гамаля заснована на проблемі дискретного логарифма [9].

Відповідна одностороння функція є показниковою функцією за простим модулем p: секретні параметри входять в показники степенів.

Піднесення числа до степеня в скінченному полі виконується легко, тоді як відновлення показника степеня за значенням функції (тобто знаходження дискретного логарифма) при великих p є складною обчислювальною задачею.

Особливістю цієї криптосистеми є те, що дискретний логарифм не є односторонньою функцією з секретом.

Тому для її обернення відправник повідомлення формує додаткову інформацію на основі разового ключа, яку одержувач може використовувати для читання повідомлення захищеним від просочування інформації способом.

В криптосистемі Ель-Гамаля для побудови пари асиметричних ключів вибирається велике просте число p і два псевдовипадкові числа, менші $p-1$.

Одне з них, g, повинно бути елементом великого порядку за модулем p, скажімо, первісним коренем. Друге число, x, вибирається як секретний ключ. Вважається, що повідомлення – лишки за модулем p.

Відкритим ключем є трійка чисел $p,q,y=g^x \bmod p$.

Для кожного повідомлення формуються додаткові дані, що грають роль лазівки для конкретного сеансу шифрування.

Для зашифрування повідомлення m вибирається псевдовипадкове число k (рандомізатор, разовий ключ) з умовою НСД $k,(p-1)=1$.

Рандомізатори не повинні повторюватися і повинні триматися в секреті.

Потім обчислюються числа $a=g^k \bmod p$ – лазівка і $b=y^x m \bmod p$ – шифртекст. Криптограмою є пара блоків даних a,b.

Для розшифрування достатньо одержати співмножник y^k, що можна зробити за допомогою секретного ключа, обчисливши значення $a^x=g^{kx} \bmod p$.

Дійсно, $y^k=g^{xk} \bmod p$, тому $m=a^{-x}b \bmod p$.

Таким чином, забезпечується захищений інформаційний обмін без попереднього розсилання секретних ключів.

2.4 Криптосистеми на основі еліптичних кривих

Для побудови СВК можуть бути використані еліптичні криві – математичні об'єкти, визначені над скінченними полями [10].

Для випадків характеристик поля $p=2$, $p=3$, $p>3$ алгоритми обчислення значень, пов'язаних з реалізацією подібних СВК, істотно різні.

Разом з тим, основні принципи, на основі яких побудовані СВК на еліптичних кривих, в ідейному значенні однакові.

Слід відзначити, що, на відміну від розглянутих раніше криптосистем, побудова відкритих параметрів для систем на еліптичних кривих є складною алгоритмічною проблемою.

В даному випадку ефект від стандартизації криптоалгоритмів особливо відчутний.

Для спрощення викладу розглянемо еліптичні криві над простим полем характеристики $p>3$.

Еліптична крива (ЕК) над полем лишків за модулем $p>3$ безпосередньо пов'язана з розв'язками рівняння $y^2=x^3+ax+b \bmod p$ за так званої умови невиродженості кривої $4a^3+27b^2\neq 0$.

Вказане порівняння називається рівнянням кривої в афінних координатах.

Кожний розв'язок (x, y) рівняння $y^2=x^3+ax+b \bmod p$ називається точкою кривої $P(x,y)$.

Множину розв'язків рівняння $y^2=x^3+ax+b \bmod p$ можна розширити таким чином, що розширена множина стане комутативною групою E.

Ця група називається групою точок на еліптичній кривій.

Груповий закон в групі E називається додаванням.

Основною причиною, що дозволяє побудувати групу E, є можливість побудови нових рішень рівняння кривої, виходячи з вже відомих.

Виявляється, якщо дані розв'язки $P_1(x_1,y_1)$ і, $P_2(x_2,y_2)$ то «практично завжди» можна знайти третій розв'язок, використовуючи знання координат перших двох.

Операція, що зіставляє двом (можливо, однаковим) точкам їх «суму», в афінних координатах записується у вигляді дробових виразів, тому при обчисленнях може виникнути особливість, якщо у відповідному знаменнику з'явиться нульове значення (за модулем p).

Очевидно, це єдина ситуація, коли виникає особливість.

Отже, їй можна зіставити деяке позначення (O) і розширити множину розв'язків рівняння кривої, додавши символ O, імітуючи тим самим існування додаткового елемента, званого нескінченно віддаленою точкою.

Можна показати, що якщо для операції «+» вважати O нейтральним елементом, то розширена множина точок кривої перетворюється на комутативну групу, а сама операція – в груповий закон.

Насправді, для аналітичного опису розширеної множини точок кривої слід використовувати так звані проективні координати, в яких точки кривої будуть задаватися не парою, а трійкою чисел.

Виявляється, в цьому випадку всі точки кривої будуть рівноправними і будуть підкорятися більш загальному груповому закону.

Груповий закон в афінних координатах був відкритий при дослідженнях еліптичних кривих на звичайному полі дійсних чисел, що «підказало» вид відповідних перетворень над скінченними полями.

Для цього слід враховувати, що якщо пряма перетинає ЕК в двох точках розширеної кривої, то вона перетинає криву в третій точці (точка дотику вважається подвійною точкою).

Для складання точок P і Q кривої (рис. 7.2, б) проводимо через P і Q пряму. Вона перетне ЕК в деякій третій точці $\tilde{R}$.

Проведемо через $\tilde{R}$ пряму, паралельну осі ординат, яка перетне ЕК в деякій точці R, яку приймемо за суму точок кривої: $R=P+Q$.

Якщо $P+Q$, то точка R є перетином кривої і дотичної до кривої в точці P. У відповідних випадках вважаємо, що прямі, паралельні осі ординат, проходять через нескінченно віддалену точку O.

Вказана геометрична побудова називається методом січних і дотичних.

Нехай $P=(u,v)$. Позначимо $\tilde{P}=(u,-v)$. Очевидно, що при нашій побудові $R=(x,y)$, а $\tilde{R}=(x,-y)$. Розглянемо точку $R+\tilde{P}$. Користуючись геометричною побудовою, легко прослідкувати, що $R+\tilde{P}=Q$. Іншими словами, додавання до точки $R=P+Q$ точки $\tilde{P}$ еквівалентно «відніманню» точки P з точки Q.

Таким чином, для визначення операції віднімання, оберненої додаванню, приймемо $(-P(u,v))=P(u,v)$.

Нехай $G_1=G$, $G_2=G+G$, і т. д. Виходячи з точки G, можна побудувати послідовність точок $G_1,\ldots,G_{n-1},G_n=O$ деякої довжини n.

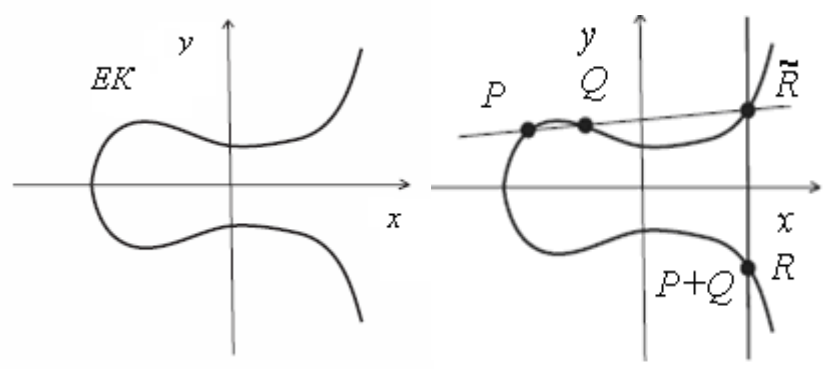
Якщо записувати подібне k-кратне додавання на кривій у вигляді kG, прийнявши $0G=O$, то, очевидно, коефіцієнт k можна приводити за модулем n і розглядати вирази вигляду $uP+vQ$ і $u(vP)$.

Операція kG називається скалярним множенням точки на k. Найменше ціле n, для якого $nG=O$, називається порядком точки G.

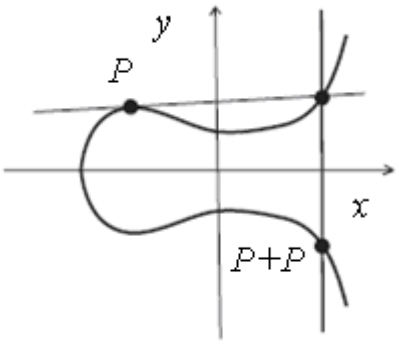
Додавання двох точок ЕК над полем дійсних чисел можна сформулювати у вигляді геометричної побудови, що зв’язує координати точок-доданків з координатами точки-суми (рис. 2.2).

Як правило, при побудові СВК на еліптичній кривій використовується точка G великого простого порядку n.

Всі операції в СВК здійснюються над точками вигляду kG, які утворюють циклічну підгрупу групи E.



а) б)



в)

Рисунок 2.2 - Графік еліптичної кривої $y^2=x^3-3x+4$ (а), додавання двох точок (б), кратна точка (в)

Використовуючи аналітичну геометрію, можна показати, що груповий закон $P_1(x_1,y_1)+P_2(x_2,y_2)=P_3(x_3,y_3)$ відповідає деяким правилам, які, у разі скінченного поля характеристики $p>3$, зводяться до нижчевикладеного.

1. $x_3=\lambda^2-x_1-x^2 \bmod p$, $y_3=\lambda(x_1-x_3) \bmod p$, де

$$\lambda=\frac{y_2-y_1}{x_2-x_1} \bmod p, \text{ якщо } P_1\neq P_2 \text{ і } \lambda=\frac{3x_1^2+a}{2y_1} \bmod p, \text{ якщо } P_1=P_2.$$

2. Якщо знаменник λ перетворюється в нуль, то $P_1(x_1,y_1)+P_2(x_2,y_2)=O$.

3. Операція, обернена додаванню: $(-P(x,y))=P(x,y)$.

4. Для будь-якої точки P розширеної кривої: $O \pm P = P$.

Покажемо, як можна побудувати аналог алгоритму Діффі-Хеллмана на еліптичній кривій.

Вихідними параметрами є сама крива E над скінченним полем і точка G великого простого порядку n.

Користувачі криптосистеми незалежно генерують секретні великі числа m_a і m_b, які є параметрами для формування спільного секретного ключа.

На їх основі обчислюються відкриті ключі $Q_a = m_a G$ і $Q_b = m_b G$.

Провівши обмін відкритими ключами, кожний з абонентів може сформувати загальний ключ $k_{ab} = m_a m_b G = m_a Q_{b=} m_b Q_a$.

Стійкість СВК заснована на складності задачі дискретного логарифмування на кривій. Ця задача полягає в знаходженні скалярного множника із співвідношення $P = kG$ і, в загальному випадку, є обчислювано недоступною.

Найбільш трудомістким етапом обчислення відкритих параметрів еліптичної кривої є визначення простого числа n, яке є дільником кількості $\#E$ елементів в групі E.

Як правило, обчислення n зводиться до обчислення $\#E$ з його подальшою факторизацією.

Для обчислення $\#E$, взагалі кажучи, використовуються складні алгоритми.

Відомі прості обмеження на коефіцієнти рівняння ЕК, при яких або $\#E = n$, або факторизація $\#E$ здійсненна.

Теоретично група E складається з не більш, ніж двох циклічних підгруп. Зазвичай параметри кривих такі, що порядок n_2 однієї з підгуп дуже малий і визначається порівняно легко.

Після чого n обчислюється у вигляді $n = \#E : n$.

Спільні алгоритми обчислення $\#E$ існують, але є складними і трудомісткими.

Для числа $\#E$ є оцінки.

Наприклад, нерівність Хассе: $q + 1 - 2\sqrt{q} \le \#E \le q + 1 + 2\sqrt{q}$, де $q = p^r$ – кількість елементів в полі характеристики p.

Таким чином, $\#E = q + 1 - t$, де $|t| \le 2\sqrt{q}$.

Для деяких кривих число $\#E$ можна обчислити теоретично.

Криві, для яких $\#E = p$, називаються аномальними, а криві, для яких $t = O(p)$, – суперсингулярними.

Для кривих цих типів відомі ефективні методи дискретного логарифмування.

Проте існує клас криптопротоколів, в яких істотно використовуються властивості саме суперсингулярних еліптичних кривих.

Слід відзначити, що багато криптографічних систем, розроблених на основі системи RSA, породжують аналоги на еліптичних кривих над кільцем лишків *Z/nZ* для складеного числа, де *p,q* — різні прості числа.

Еліптична крива *En* над *Z/nZ* задається тими ж алгебраїчними рівняннями, що і у разі простого модуля.

Наприклад, крива задається як множина розв'язків основного рівняння $y^2=x^3+ax+b \bmod n$, при НСД $(4a^3+27b^2,n)=1$ з нескінченно віддаленою точкою *O*.

Операції на *En* виконуються за формулами, відповідними методу січних і дотичних. На жаль, такий об'єкт не є групою.

Тому замість *En* розглядається так звана пряма сума кривих $\tilde{E}_n = E_p + E_q$: інший об'єкт, який «мало» відрізняється від *En*.

Кожний елемент $\tilde{E}_n$ – чотиривимірний вектор, що складається з двох точок кривих *Ep* і *Eq*.

Якщо пара *(x,y)* задовольняє основне рівняння для *En*, то пари залишків ($x \bmod p$, $y \bmod p$) і ($x \bmod q$, $y \bmod q$) автоматично задовольняють те ж рівняння за модулями *p* і *q* відповідно.

Нагадаємо, що Китайською теоремою про лишки називається наведене нижче твердження.

Нехай числа $m_1,m_2,\ldots,m_k$ попарно взаємно прості і $M= m_1m_2\cdots m_k$.

Тоді єдиний за модулем *M* розв'язок системи порівнянь $x=c_i \bmod m_i$, $i=1,\ldots k$ має вигляд:

$$x = \sum_{i=1}^{k} c_i M_i N_i \bmod M \ , \qquad (2.1)$$

де $M_i=m_1m_2\cdots m_{i-1}m_{i+1}\cdots mk$, $N_i=M_i^{-1} \bmod m_i$.

Дійсно, у вказаному виразі для x доданок c_i,M_i,N_i порівнянний з c_i за модулем m_i, а всі інші порівнянні за цим модулем з нулем.

Враховуючи Китайську теорему про лишки, *x* і *y* однозначно відновлюються за значеннями ($x \bmod p$, $y \bmod p$) і ($x \bmod q$, $y \bmod q$).

Проте, разом з нескінченно віддаленою точкою O, група $\tilde{\tilde{E}}_n$ містить сукупність точок виду *{(Op,y),(x,Oq)}*.

Тому $\tilde{\tilde{E}}_n$ складається з точок кривої E_n і деякої множини особливих точок.

Наведемо основні властивості $\tilde{\tilde{E}}_n$, близькі до властивостей E_n, що дозволяє будувати RSA-подібні криптосистеми.

1. Метод січних і дотичних у разі, коли він визначений для E_n, збігається з груповою операцією в $\tilde{\tilde{E}}_n$.

Таким чином, якщо виконується додавання конкретних точок в E_n, то воно виконується без факторизації *n*.

2. При великих *p* і q з ймовірністю, близькою до одиниці, множення точки на скаляр в $\tilde{\tilde{E}}_n$ поводиться як аналогічна операція в E_n.

Суть цього твердження така.

Приймемо *N*=НСД*(#Ep,Eq)*. Тоді (kn+1)=P для переважного числа точок і будь-яких цілих *k*.

Розглянемо аналог системи RSA з використанням еліптичної кривої.

Відкритим ключем є пара *N,e*, де *N*=НСД*(#Ep,Eq)*, *(N,e)=1*, секретним – число *d* таке, що *ed=1modN*.

Шифртекст повідомлення *m*, яке зіставляється наперед вибраним способом деякій точці M кривої *En*, Відправник одержує з допомогою скалярного множення у вигляді C=eM.

Одержувач, знаючи число *d*, відновлює *m=deM=dC*.

Виявляється, однак, що зіставлення *m↔M* може вимагати факторизацію *n*, тобто вказана схема шифрування, взагалі кажучи, не є коректною.

Проте існує ряд більш досконалих RSA-подібних криптосистем, наприклад, криптосистема Демітко (для кривих загального вигляду), або криптосистема Кувакадо-Кояма для кривих вигляду $y^2=x^3+b \bmod n$.

Еліптичні криві з параметрами, що визначені над скінченними полями мають ряд істотних переваг у порівнянні з тими, що визначені над раціональними числами, а саме – теоретична ефективність реалізації в обчислювальній техніці.

Це обумовлено тим, що у випадку поля GF(p) , числа можуть бути легко поданими у вигляді бітових слів, адже обчислювальні пристрої оперують саме з числами у двійковому поданні, а операція обчислення за модулем може бути представлена як звичайна булева функція XOR. Для поля GF(2m), многочлени, що є елементами поля можуть бути

представлені у вигляді кодових двійкових слів, де кожен розряд позначає коефіцієнт при доданку-ступені. Наприклад, многочлен $\alpha + \alpha + 1$ може бути представлений на апаратному рівні у вигляді кодового слова «1011» або $1*\alpha + 1*\alpha + 0*\alpha + 1*\alpha = \alpha + \alpha + 1$.

Дане представлення дозволяє оперувати елементами поля як звичайними цілочисельними значеннями, коли додавання, віднімання і т.д. виконуються за модулем 2, тобто використовуючи ту саму булеву логічну функцію XOR.

Що стосується алгоритмів, що необхідні для знаходження елементів поля та виконання арифметичних операцій над ними, як наприклад, алгоритм Евкліда, то даний алгоритм є за своєю природою ітераційним, що означає можливість його ефективної програмної реалізації для електронно-обчислювальних пристроїв.

Що стосується програмної реалізації операцій над точками еліптичної кривої, то слід зазначити, що операції над точками еліптичної кривої є за своєю суттю комбінацією арифметичних операцій над параметрами еліптичної кривої та координатами її точок.

Таким чином еліптична крива в обчислювальній техніці може бути однозначно представлена парою цілочисельних значень координат (x; y). Окремо слід зазначити випадок операцій за участю так званої точки на нескінченності.

Складність даного випадку полягає у тому, що точка на нескінченності є скоріше математичною концепцією, адже у цієї точки невизначені координати (нескінченність), що очевидно не належать скінченному полю, над яким визначена дана крива. Таким чином, результати операцій за участю точки на нескінченності повинні бути оброблені як граничні випадки, тобто аксіоматично визначеними в тілі алгоритму, а саме:

1. $O = -O$
2. $O + P = P$
3. $k \cdot O = O$

З точки зору об’єктного проектування, операції над точками еліптичної кривої є за своєю суттю надбудовою над операціями в скінченних полях.

2.5 Висновки до розділу 2

У другому розділі магістерської роботи проведено огляд та дослідження алгоритмів та моделей криптосистеми з відкритими ключами.

Серед характеристик криптосистеми з відкритими ключами визначено, що найслабшою ланкою при реалізації симетричних криптосистем в системах захищеного електронного документообігу, електронних банківських платежів і, особливо, електронної торгівлі є питання розподілу ключів.

Для забезпечення обміну конфіденційною інформацією між двома абонентами телекомунікаційної мережі повинен бути згенерований ключ (можливо, одним з абонентів), а потім по деякому захищеному каналу переданий іншим користувачам (іншому абоненту). Як засіб створення захищеного каналу знову може бути використана деяка асиметрична криптосистема.

Найбільшу гостроту питання розподілу і доставки ключів набуває у разі неможливості наперед описати склад інформаційно-телекомунікаційної мережі.

В СВК для зашифрування не використовуються секретні ключі – вони необхідні тільки при розшифрування.

Таким чином, одностороння функція з секретом, що використовується в асиметричній системі, є взаємооднозначною, але разом з тим має властивості необерненості.

Розробники криптосистеми RSA скористалися тим фактом, що знаходження великих простих чисел в обчислювальному відношенні здійснюється достатньо просто, але невідомий алгоритм, що виконує за поліноміальний час розкладання на прості множники великих чисел.

Доведено (теорема Рабіна), що розкриття шифру RSA еквівалентне знаходженню такого розкладу. На відміну від RSA асиметрична криптосистема Ель-Гамаля заснована на проблемі дискретного логарифма.

Для побудови СВК можуть бути використані еліптичні криві – математичні об'єкти, визначені над скінченними полями.

Слід відзначити, що багато криптографічних систем, розроблених на основі системи RSA, породжують аналоги на еліптичних кривих.

Визначені шляхи подальших досліджень програмних засобів створення ЕЦП.

2.6 Перелік джерел посилань до розділу 2

1. Розподіл ключів. Безпека та захист інформації. [Електронний ресурс]. – Режим доступу: https://iconfs.net/s.infocom2018/rozpodil-klyuchiv (дата звернення 24.10.2020).

2. Криптографія з відкритим ключем, різновидності, алгоритм. [Електронний ресурс]. – Режим доступу: https://sites.google.com/view/blog-ua/основні-поняття-криптографії-та-захисту-інформації/криптографія-з-відкритим-ключем-різновидності-алгоритм (дата звернення 24.10.2020).

3. Кравцов Г.О. Однонаправлена функція. Базові визначення та теореми. [Електронний ресурс]. – Режим доступу: http://www.itsway.kiev.ua/pdf/Articles06022006s.pdf (дата звернення 24.10.2020).

4. Как работает обмен ключами в протоколе Диффи-Хеллмана. [Електронний ресурс]. – Режим доступу: https://tproger.ru/translations/diffie-hellman-key-exchange-explained (дата звернення 24.10.2020).

5. Ранцевая криптосистема Меркла- Хеллмана. [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Ранцевая_криптосистема_Меркла_—_Хеллмана (дата звернення 24.10.2020).

6. Криптосистем з відкритими ключами. [Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5367441/ (дата звернення 24.10.2020).

7. Шифр Эль-Гамаля. [Електронний ресурс]. – Режим доступу: https://it.rfei.ru/course/~k017/~V8u3Fj4l/~hIGNMjZS (дата звернення 24.10.2020).

8. Ю. Подолюк, А. Переймибіда. Захист інформації в інтернет-застосуваннях на основі криптосистем з еліптичними кривими. [Електронний ресурс]. – Режим доступу: visnyk-ami.lnu.edu.ua (дата звернення 24.10.2020).

9. Жалобкевич Е.В., Липницкий В.А. Подходы к решению проблемы дискретного логарифмирования. [Електронний ресурс]. – Режим доступу: https://elib.bsu.by/bitstream/123456789/156465/1/6-8.pdf (дата звернення 24.10.2020).

10. Дичка А.І. Модифікація методу багатократного скалярного множення точок еліптичної кривої на число [Текст] / М.В. Онай, Дичка А.І. // Прикладна математика та комп'ютинг. ПМК, 2014 : десята наук. конф. магістрантів та аспірантів, Київ, 16-18 квітня 2014 р. : зб. тез доп. / [ред кол.: Дичка І.А. та ін.] . – К. : Просвіта, 2018. – С. 235-240.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ТЕСТУВАННЯ ЧИСЕЛ НА ПРОСТОТУ І ВИБІР ПАРАМЕТРІВ RSA

При побудові асиметричних криптосистем, а також модифікації з параметрів в ході експлуатації виникає необхідність побудови надвеликих псевдовипадкових простих чисел, що мають ті або інші специфічні властивості.

У багатьох випадках, наприклад, у випадку RSA, великі прості числа є ключовими параметрами.

Відповідні обчислювальні процедури включають в себе алгоритми, що реалізують етап перевірки чисел на простоту.

В літературі і криптографічній практиці подібні алгоритми носять назву тестів.

3.1 Детерміновані тести при побудові асиметричних криптосистем створення ЕЦП

В основі тестів лежать так звані критерії простоти.

Існує два типи критеріїв простоти: детерміновані і ймовірнісні [1].

Детерміновані тести дозволяють довести, що число, яке тестується, – просте. Практично застосовувані детерміновані тести здатні дати позитивну відповідь не для кожного простого числа, оскільки використовують лише достатні умови простоти.

Детерміновані тести більш корисні, коли необхідно побудувати випадкове велике просте число, а не перевірити простоту, скажімо, деякого єдиного числа.

Детермінований тест використовується, наприклад, в процедурах обчислення несекретних параметрів цифрового підпису типу Ель-Гамаля, встановлених ГОСТ 34.310. Цей тест заснований на викладеній нижче теоремі.

3.2 Теорема Демітко

Нехай $n = qR + 1$, де q – просте, R – парне і $R < 4(q+1)$ [2].

Якщо існує а таке, що $a^{n-1} \equiv 1 \bmod n$ і $a^{(n-1)/q} \neq 1 \bmod n$, то число n – просте.

На відміну від детермінованих, імовірнісні тести можна ефективно використовувати для тестування окремих чисел, проте їх результати, з деякою ймовірністю, можуть бути недостовірними.

На щастя, ціною кількості повторень тесту з модифікованими вихідними даними ймовірність помилки можна зробити як завгодно малою.

3.3 Тест на основі малої теореми Ферма

При побудові ймовірнісних критеріїв простоти виникає ряд типових питань, які зручно розглянути на прикладі, за який виберемо тест на основі малої теореми Ферма.

Як відомо, ця теорема стверджує, що якщо n – просте, то для всіх a, взаємно простих з n, виконується умова (порівняння Ферма): $a^{n-1} \equiv 1 \bmod n$. [3].

Таким чином, якщо порівняння Ферма не виконано хоча б для одного числа з множини $\{2,\ldots,n1\}$, то a – складене.

Тест на основі малої теореми Ферма пояснюється нижче.

Псевдовипадково вибираємо лишок $a \in \{2,\ldots,n-1\}$ і перевіряємо умову $(a,n)=1$. Якщо ця умова не виконана, значить, n – складене. Перевіряємо порівняння Ферма.

Якщо воно не виконується, то число n – складене. Інакше, повторюємо тест для іншого значення a.

Очевидно, основне питання полягає в тому, щоб оцінити, якою мірою тест є ефективним. Перш за все, слід з'ясувати, чи існують складені числа n, для яких при деякому a виконується умова малої теореми Ферма.

Виявляється, це так, оскільки існують контрприклади: $2^{340}=(2^{10})^{34}=1 \bmod 341$, $7^{24}=1 \bmod 25$.

Назвемо складене непарне число n псевдопростим за основою a, якщо пара чисел задовольняє порівняння Ферма $a^{n-1} \equiv 1 \bmod n$.

Виявляється, можна показати, що для будь-кого a існує нескінченно багато псевдопростих чисел за основою a.

3.4 Основні властивості псевдопростих чисел

Основні властивості псевдопростих чисел такі [4].

Нехай n – непарне складене число. Тоді:

1) n - псевдопросте за основою a в тому і лише тому випадку, коли $(a,n)=1$ і $ord_n a \mid n-1$;

2) якщо n – псевдопросте за основою a і b, то n – псевдопросте за основами ab і $ab^{-1} \bmod n$;

3) множина $F_n = \{a \in Z_n : a^{n-1} \equiv 1 \bmod n\}$ утворює мультиплікативну підгрупу в мультиплікативній групі Z_n^* обернених елементів кільця лишків за модулем n;

4) якщо n не є псевдопростим за основою a хоча б для одного числа a, то $|F_n| \le (1/2)|Z_n^*|$ (тут через $|F|$ позначено кількість елементів, що належать множині F).

Помітимо, що $|Z_n^*| = \varphi(n) \le n-1$. Таким чином, якщо тест Ферма виявляє складене n при одній основі a, то існує не менше $(n-1)/2$ основ з аналогічною властивістю.

Дійсно, властивість 1) виходить з визначення $ord_n a$. Властивості 2) і 3) виходять з правила почленного перемножування частин порівнянь і перевірки порівняння Ферма для основи b^{-1}.

Доведемо властивість 4). Нехай $F_n = \{a_1, \ldots, a_s\}$ і a – основа, за якою n не є псевдопростим. Тоді для будь-якого i пари чисел aa_i, n не задовольняють порівняння Ферма.

Тому кількість основ, для яких n не є псевдопростим, не менше, ніж кількість елементів в F_n.

Отже, якщо $|F_n| > (1/2)|Z_n^*|$, то спільне число елементів у Z_n^* виявиться більше n, що неможливо.

Таким чином, можна сказати, що якщо існує (навіть не відома нам) основа, за якою n не є псевдопростим, то, при повторенні тесту Ферма k раз, ймовірність k-кратного вибору основ з множини F не перевищує $(1/2)^k$.

В цьому випадку ймовірність помилки тесту наближається до нуля зі збільшенням k.

Проблема може виникнути лише в тому випадку, якщо n є псевдопростим для всіх (ненульових) основ.

Виявляється, такі числа існують. Вони називаються числами Кармайкла. Наприклад, числом Кармайкла є число $n = 561 = 3 \cdot 11 \cdot 17$.

3.5 Властивості чисел Кармайкла

Властивості чисел Кармайкла описуються такою теоремою [5].

Нехай n – непарне складене число. Тоді якщо:

1) $p^2 | n$, $p > 1$, то n не є числом Кармайкла;

2) $n = p_1 p_2 \cdots p_k$, $p_i \neq p_j$ для $(i \neq j)$, то n – число Кармайкла в тому і лише тому випадку, коли $\forall i \ (p_i - 1) \vdots (n-1)$;

3) $n = p_1 p_2 \cdots p_k$, $p_i \neq p_j$ для $(i \neq j)$ – число Кармайкла, то $k \geq 3$.

Числа Кармайкла є достатньо рідкісними. В межах до 100000 існує лише 16 чисел Кармайкла: 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657, 52633, 62745, 63973, 75361.

З попереднього випливає, що при тестуванні чисел на простоту за допомогою імовірнісного тесту, заснованого на малій теоремі Ферма, може виникнути ситуація, коли ймовірність помилки не знижується з кількістю повторень тесту.

В подібному випадку згадана ймовірність рівна одиниці, і в результаті тестування може бути прийнято неправильне рішення.

В зв'язку з цим розроблені і застосовуються на практиці імовірнісні тести, вільні від вказаного недоліку.

Прикладами таких тестів є тест Соловея-Штрассена [6] і тест Рабіна-Міллера [7] перевірки чисел на простоту.

3.6 Тест Соловея-Штрассена і Ейлерові псевдопрості числа

В тесті Соловея-Штрассена використовуються властивості так званих символів Лежандра і Якобі, пов'язаних з розв’язністю двочленних квадратичних порівнянь. Нагадаємо коротко їх властивості.

Двочленним квадратичним порівнянням називається порівняння виду $x^2 = a \bmod n$, де x - невідомий лишок.

Ціле число a називається квадратичним лишком за модулем n, якщо порівняння $x^2 = a \bmod n$ має розв’язки.

Якщо порівняння має розв’язки, то для складеного непарного модуля кількість розв’язків, як правило, більша двох.

В загальному випадку, не тільки розв’язання квадратичних порівнянь, але навіть питання про розв’язність двочленного квадратичного порівняння за складеним модулем, факторизація якого невідома, є алгоритмічною проблемою, на складності якої заснований ряд криптопротоколів і криптосистем.

В той же час для модулів, що є простими числами, задача легко піддається аналізу.

Існує ефективний алгоритм для визначення є дане число квадратичним лишком за простим непарним модулем p, чи ні. Цей алгоритм дозволяє обчислити, практично вручну,

так званий символ Лежандра $\left(\frac{a}{p}\right)$, значення якого при $a \neq 0$ рівно 1, якщо a - квадратичний лишок, і (-1) – в іншому випадку. Для $a \equiv 0 \bmod p$ вважається $\left(\frac{a}{p}\right) = 0$.

З появою комп'ютерів значення $\left(\frac{a}{p}\right)$ звичайно обчислюється, виходячи із співвідношення: $\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p$, яке є важливою властивістю символу Лежандра.

Якщо непарне число n факторизовано: $n = \prod_{i=1}^{k} p_i^{a_i}$, то розв'язання порівняння $x^2 = a \bmod n$ еквівалентне розв'язанню всіх порівнянь виду $x^2 = a \bmod p_i^{a_i}$.

Помітимо, що, у свою чергу, порівняння $x^2 = a \bmod p_i^{a_i}$ має корені тоді і тільки тоді, коли $\left(\frac{a}{p_i}\right) = 1$, а також, добуток двох квадратичних нелишків є квадратичним лишком за простим модулем.

Тому значення $\left(\frac{a}{p}\right)$ можна обчислити через величини $\left(\frac{a}{p_i}\right)$. З символом Лежандра тісно пов'язаний символ Якобі $\left(\frac{x}{n}\right)$ числа x за модулем n.

Символ Якобі визначається для непарних, взаємно простих чисел як добуток значень символів Лежандра: $\left(\frac{x}{n}\right) = \left(\frac{x}{p_1}\right)^{a_1} \cdots \left(\frac{x}{p_k}\right)^{a_k}$.

Він має практично всі ті ж властивості, що і символ Лежандра, але за значенням символу Якобі, рівним одиниці, не можна стверджувати, що відповідний лишок x – квадратичний. Для квадратичного лишку, проте, символ Якобі рівний одиниці.

Отже, якщо $\left(\frac{x}{n}\right) = -1$, то x - квадратичний нелишок за модулем n.

Ця особливість пов'язана з тим, що співвідношення $\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p$ не обов'язково виконується для символу Якобі, тобто, коли числор (модуль) не є простим. Для нас символ Якобі важливий тому, що його можна обчислити без факторизації модуля.

На основі нижченаведеної теореми, в тесті Соловея-Штрассена використовується критерій Ейлера для визначення значення символу Лежандра (квадратичного характеру

числа за простим модулем). В самому тесті при тестуванні числа n обчислюється символ Якобі $\left(\frac{a}{n}\right)$.

Теорема [17]. Непарне ціле число $n>1$ є простим тоді і тільки тоді, коли для всіх чисел a: $1\le a\le n-1$ виконується співвідношення Ейлера: $a^{\frac{n-1}{2}}=\left(\frac{a}{n}\right)\bmod n$.

Складене число n, що задовольняє співвідношення Ейлера, називається ейлеровим псевдопростим за основою a.

З вказаної теореми виходить, що складених чисел, які були б ейлеровими псевдопростими за будь-якою основою, не існує.

Отже, ми можемо запропонувати такий тест, аналогічний тесту Ферма.

Псевдовипадково вибираємо залишок $a\in\{2,\ldots,n-1\}$ і перевіряємо умову $(a,n)=1$. Якщо умова не виконана, значить, n – складене.

Перевіряємо співвідношення Ейлера.

Якщо воно не виконується, то число n – складене.

Інакше, повторюємо тест для іншого значення a.

Якщо ми могли б перевірити співвідношення Ейлера для всіх значень a, то ми змогли б точно визначити, чи є число n простим.

Але для великих n це неможливо.

Тому необхідно оцінити, як веде себе ймовірність помилки при збільшенні числа k повторень тесту.

Це можна зробити, виходячи з твердження, аналогічного тому, яке ми розглядали при аналізі властивостей псевдопростих чисел.

Цікаво, що ейлерові псевдопрості є псевдопростими числами.

Теорема [17]. Нехай n – непарне складене число. Тоді:

а) якщо n – ейлерове псевдопросте за основою a, то воно – псевдопросте за основою a;

б) якщо n – ейлерове псевдопросте за основами a і b, то n – ейлерове псевдопросте за основами ab і $ab^{-1}\bmod n$;

в) множина $E_n=\left\{a\in Z_n : a^{(n-1)/2}=\left(\frac{a}{n}\right)\bmod n\right\}$ є підгрупою групи $F_n=\{a\in Z_n : a^{n-1}=1\bmod n\}$;

г) якщо n не є ейлеровим псевдопростим за основою a хоча б для одного числа a, то $|E_n| \le (1/2)|Z_n^*|$.

Таким чином, при повторенні тесту Соловея-Штрассена k раз ймовірність невідбракування складеного числа не перевершує $(1/2)^k$.

3.7 Тест Рабіна-Міллера і сильні псевдопрості числа

Ще більш ефективним ймовірнісним тестом є тест Рабіна-Міллера, в якому використовується критерій, в кінцевому рахунку, оснований на факті, що для простого модуля квадратними коренями з одиниці є лише числа ±1, а для складеного непарного модуля *n=uv, (u,v)=1,* число таких коренів більше двох.

Нехай *n* – непарне натуральне число. Тоді можна записати $n-1=2^s t$, де t – непарне і $s \ge 1$.

Якщо число n – просте, то a^{n-1}=1mod*n*, при *(a,n)=1*. Тому квадратні корені з одиниці мають вигляд: $a^{(n-1)/2}$=(±1)modn., де показник рівний $2^{s-1}t$.

Це означає, що в послідовності $a^t, a^{2t}, \ldots, a^{2^{s-1}t}$ лишків за простим модулем, які є послідовними квадратами числа a^t, або з'явиться (-1)modn, або всі ці лишки порівнянні з одиницею, тобто a^t=1modn. Помітимо, що при простому n лівіше (-1)mod*n* можуть розташовуватися лише лишки не рівні (±1)mod*n*.

Якщо n – складене, то можливі й інші варіанти, оскільки в цьому випадку крім ± 1 існують інші корені з одиниці за модулем *n*.

Заснований на даному зауваженні тест Рабіна-Міллера полягає в тому, що:

1) псевдовипадково вибираємо лишок $a \in \{2, \ldots, n-1\}$ і перевіряємо умову *(a,n)=1*. Якщо умова не виконана, значить, n – складене і робота закінчена;

2) обчислюємо a^tmodn. Якщо a^t=(±1)modn, то не виключено, що число n – просте і необхідно перейти на початок, щоб повторити тест для іншої основи;

3) обчислюємо послідовно лишки чисел $a^{2t}, \cdots a^{2^{s-1}t}$ за модулем n, поки не з'явиться (-1), або не вичерпається список;

4) якщо (-1) знайдено в списку, то не виключено, що число n – просте і необхідно перейти на початок, щоб повторити тест для іншої основи;

5) якщо жодне число із списку не порівнянно з (-1), то число n – складене і необхідно закінчити роботу.

Як і для інших імовірнісних тестів, існують складені числа n, які, для відповідних основ a, проходять даний тест.

Назвемо число $n = 2^s t + 1$, де $s \geq 1$, t – непарне, сильним псевдопростим за основою $a \neq 1 \bmod n$, (a,n)=1, якщо виконується одна з двох умов: $a^t = \pm 1 \bmod_n$, або в послідовності існує число, порівнянне з -1 за модулем n.

Виявляється, можна показати, що для будь-якого a, (a,n)=1), існує нескінченно багато сильних псевдопростих чисел n за основою a.

Приклади: a=7, n=25, a=5, n=781;.

Можна довести такі основні властивості сильних псевдопростих чисел:

1) число n, сильне псевдопросте за основою a, є ейлеровим псевдопростим за тією ж основою;

2) якщо непарне складене число n є сильним псевдопростим за основою a, то загальна кількість основ, за якою це число є сильним псевдопростим, не перевищує (n-1)/4.

Тому можна стверджувати, що при повторенні випробувань тесту Рабіна-Міллера k раз ймовірність невідбракування складеного числа $\leq (1/4)^k$.

Крім того, виявляється, кількість повторень тесту, достатню для практичних додатків, можна обмежити величиною $2\log_2^2 n$.

Цікаво відзначити, що простоту невеликих простих чисел можна довести, використовуючи декілька раніше вказаних основ.

Приклади [17]: якщо $n < 1373653$ – сильне псевдопросте за основами 2 і 3, то n – просте; якщо $n < 341550071728321$ – сильне псевдопросте за основами, 3, 5, 7, 11, 13, 17, то n – просте.

Загальні вимоги до вибору параметрів RSA.

Коректність параметрів RSA пов'язана з оцінюванням стійкості системи і може бути визначена лише з погляду практичної стійкості.

Отже, коректні параметри повинні бути побудовані так, щоб мінімізувати шкоду від відомих підходів до ослаблення криптосистеми.

Виходячи з цих міркувань, розглянемо найбільш загальні вимоги до вибору параметрів *p,q,e,d* криптосистеми RSA [17,18].

Перш за все, слід враховувати, що слабкість одного з параметрів практично не компенсується посиленням властивостей інших параметрів.

Очевидно, число $n=pq$ повинно бути великим. Числа p і q не повинні міститися в списках відомих великих простих чисел, не повинні бути дуже близькі один до одного, або істотно розрізнятися за величиною.

Вони не повинні бути побудованими за детермінованими алгоритмами з невеликим числом відомих варіантів початкових параметрів або містити закономірності в двійковому записі.

Загалом, p і q не повинні відрізнятися від типових представників випадкових простих чисел. Аналогічні властивості повинні мати параметри e і d.

Наприклад, якщо секретний ключ d містить в двійковому записі невелику кількість одиниць, то номери місць цих одиниць легко визначити перебором.

Можна довести, що при відомому d існує можливість факторизації модуля.

Відомо, що для читання повідомлень, зашифрованих криптосистемою RSA, достатньо знання деякого кратного функції Ейлера від модуля, оскільки в цьому випадку можна обчислити ключ, криптоеквівалентний ключу d.

Відзначимо також, що за наявності $ord_n a$ легко одержати a з порівняння $a^e=c \bmod n$. Достатньо піднести c до степеня h, що задовольняє співвідношення.

Для будь-якого a, взаємно простого з $n=pq$, $ord_p a$ ділить $p-1$, а $ord_q a$ ділить $q-1$. Тому $ord_n a$ ділить G=НСД((p-1),(q-1)).

Отже, для побудови криптосистеми, замість визначення d з порівняння $ed=1 \bmod \varphi(n)$, можна скористатися розв'язком порівняння $ed_1=1 \bmod G$.

Нехай $g = \text{НСД}((p-1),(q-1))$. Тоді $gG = \varphi(n)$.

Очевидно, із співвідношення $ed=1 \bmod \varphi(n)$, виходить $ed=1 \bmod G$, тому $d=d1 \pmod G$ і $d \neq d_1 (\bmod \varphi(n))$.

Ці умови задовольняють ключі d_1, d_1+G, d_1+2G, $\cdots$, $d_1+(g-1)G$, криптоеквівалентні, таким чином, ключу d. Отже, чим більше $\text{НСД}((p-1),(q-1))$, тим більше криптоеквівалентних ключів, тим гірше для криптосистеми.

Очевидно, в найкращому можливому випадку, $\text{НСД}((p-1),(q-1))=2$, при цьому, $p=2s+1$, $q=2t+1$, де $(s,t)=1$, скажімо, s і t – прості.

Відзначимо, до речі, що невідомо, чи є множина простих чисел виду $s=(p-1)/2$ нескінченною.

Виявляється, щоб уникнути можливості застосування більшості часткових методів факторизації для дешифрування криптосистеми RSA, достатньо вимагати, щоб числа

$p_1=(p-1)/2$, $p_2=(p+1)/2$, $q_1=(p-1)/2$ і $q_2=(p+1)/2$ не розкладалися в добуток степенів невеликих простих чисел, тобто щоб вони містили в розкладі велике просте число.

Відповідні вимоги, в найбільш сильній формі, полягають в тому, щоб числа p_1, p_2, q_1, q_2 були простими, більш того, вимагається, щоб в розкладі як p_1-1, так і q_1-1 було велике просте число.

На практиці, при побудові відповідного цим вимогам простого числа, скажімо p, достатньо, щоб існував достатньо великий простий дільник числа $p-1$. Очевидно, такий дільник має вигляд $r=(p-1)/2j$.

Таким чином, необхідно виділити деякий специфічний клас простих чисел.

Визначення. Просте число p називається сильним простим, якщо виконуються умови:

$p\equiv 1 \bmod r$, $p\equiv -1 \bmod s$, $r\equiv 1 \bmod t$,

де r, s, t – великі прості числа.

Оскільки числа p, r, s, t – непарні, то вони подаються у вигляді $p=1+2jr$, $p=-1+2ks$, $r=1+2lt$. Крім того, для наших цілей чим менші числа j, k, l, тим краще.

3.8 Метод Гордона побудови сильних простих чисел

Для побудови сильних простих чисел застосовується так званий метод Гордона [8].

В цьому методі здійснюється перегляд околів деяких псевдовипадкових чисел з метою виявлення простих чисел, що задовольняють специфічні умови.

При цьому неодноразово використовуються імовірнісні процедури побудови проміжних даних, а також застосовується тестування чисел на простоту за допомогою тесту Рабіна-Міллера.

Суть методу Гордона побудови сильного простого числа p така.

1. Будуємо випадкове просте число s, виходячи з заздалегідь вибраної для нього розрядності h. Для цього вибираємо псевдовипадково число x розрядності h і за допомогою методу пробних ділень залишаємо в проміжку $x, x+\log_2 x$ числа, що не мають малих дільників. Серед чисел, що залишилися, за допомогою тестів на простоту, визначаємо просте число s.

2. Будуємо просте число $t\neq s$ аналогічно побудові числа s.

3. За допомогою методу пробних ділень і тестів на простоту аналогічно пункту 1 будуємо просте число $r = 1 + 2lt$, перебираючи l в проміжку $[1, \log_2 t]$.

4. Обчислюємо $u = u(r,s) = \left(s^{r-1} - r^{s-1}\right) \bmod rs$.

Щоб не підносити до степеня, це зручно зробити за допомогою китайської теореми про лишки, оскільки $u \equiv 1 \bmod r$ і $u \equiv -1 \bmod s$.

Вимагатимемо, щоб шукане число p задовольняло ті ж умови: $p = u \equiv 1 \bmod r$ і $p = u \equiv -1 \bmod s$.

5. Якщо число u – непарне, то присвоюємо $p_0 = u$, інакше вважаємо $p_0 = u + rs$.

6. Будуємо p – найближче просте число порівнянне з непарним числом p_0 за модулем rs, тобто тестуємо на простоту числа вигляду $p = p_0 + 2krs$, $k = 0, 1, \ldots$, поки не знайдеться просте число (або спрацюють обмеження реалізації).

Алгоритм заснований на такій теоремі.

Теорема (Гордон). Якщо r, s – непарні прості числа, то просте число p задовольняє умови $p = u \equiv 1 \bmod r$ і $p = u \equiv -1 \bmod s$.

Тоді і тільки тоді, коли воно подано у вигляді $p = p_0 + 2krs$, де p_0 – непарне число з пари $u, u + rs$.

3.9 Приклад побудови сильного простого числа

1. Будуємо вихідне псевдовипадкове просте число s, розміром, скажімо, в 6 бітів. Вибираємо псевдовипадково шестибітове число: $x = 46$. В проміжку [46,46+5] визначаємо просте число $s = 47$.

2. Будуємо випадкове просте число t аналогічно побудові числа s. Нехай $x = 25$. В проміжку [25,25+4] визначаємо просте число $t = 29$.

3. Будуємо просте число $r = 1 + 2lt$, перебираючи l в проміжку [1,4]. Отримуємо $l = 59$.

4. Обчислюємо $u = u(r,s) \bmod rs$, розв'язуючи за допомогою китайської теореми про лишки систему: $u \equiv 1 \bmod r$, $u = -1 \bmod s$.

Використовуючи розширений алгоритм Евкліда, отримаємо співвідношення

$4 \cdot 59 - 5 \cdot 47 = 1$, звідки: $47^{-1} \bmod 59 = 4 = -5$.

Отже,

$$u(r,s) = 1 \cdot 47 \cdot \left(47^{-1} \bmod 59\right) + (-1)59 \cdot \left(59^{-1} \bmod 47\right) = -471 = 2302 (2773).$$

5. Число $u(r,s)$ – парне, тому вважаємо $p_0 = 2302 + 59 \cdot 47 = 5075$.

6. Будуємо просте число порівнянне з p_0 за модулем 2773, тестуючи на простоту числа вигляду $p = p_0 + 2 \cdot 2773k$.

При $k = 0, 1, 2, 3$, отримуємо, відповідно: 5075, 10621, 11167, 21713.

Лише останнє число є простим. Його розрядність значно перевищує розрядність вихідного числа s, що небажано, оскільки, наприклад, для криптосистеми RSA потрібна побудова сильних простих чисел заданої розрядності.

При великій розрядності чисел, що використовуються в обчисленнях, кількість сильних простих велика, тому вказаний вище недолік проявляється значно менше.

Проте завжди необхідно передбачати відповідні дії у випадку, якщо проміжні дані на відповідному кроці побудувати не вдалося. Крім того, слід завжди контролювати розрядність проміжних даних і результатів.

Одним з яскравих представників сімейства криптографічних алгоритмів з використанням еліптичних кривих є алгоритм електронно-цифрового підпису ECDSA.

При застосуванні криптографічних алгоритмів з використанням еліптичних кривих дуже важливим чинником є час їхньої роботи. Експериментальні дослідження показують, що найбільш ресурсо- та часовитратними є операції, що виконуються безпосередньо з точками еліптичної кривої, зокрема для алгоритму ECDSA найбільш ресурсовитратною є операція багатократного скалярного множення точок еліптичної кривої на число. Таким чином, актуальними є дослідження способів та методів оптимізації даного обчислення.

В межах криптографії, активне застосування знаходять еліптичні криві з параметрами, визначеними над скінченними полями, а саме полями з простою характеристикою – GF(p) та бінарним розширенням – GF(2^m). Це в свою чергу означає, що координати точок еліптичної кривої належать деякому скінченному полю, що знімає проблему округлення значень при виконанні операцій над ними.

В межах криптографії також розглядаються математичні основи теорії скінченних полів або полів Галуа, зокрема, полів GF(p) та GF(2^m), арифметичних дій над її елементами, арифметичні операції над точками еліптичної кривої та особливості програмної реалізації зазначених компонентів. Вводяться поняття характеристики поля, мультиплікативної групи, незвідного полінома, додавання, віднімання, множення елементів поля, знаходження оберненого елемента тощо. Для подання однозначного і вичерпного визначення скінченного поля вводяться математичні визначення, що використовуються для формулювання означення поля.

ECDSA (Elliptic Curve Digital Signature Algorithm) – алгоритм з відкритим ключем для створення цифрового підпису, аналогічний, за своєю будовою, DSA, але визначений, на відміну від нього, не над полем цілих чисел, а у групі точок ЕК.

3.10 Висновки до розділу 3

У третьому розділі магістерської роботи проведено дослідження тестування чисел на простоту і вибір параметрів алгоритму RSA.

При побудові асиметричних криптосистем, а також модифікації з параметрів в ході експлуатації виникає необхідність побудови надвеликих псевдовипадкових простих чисел, що мають ті або інші специфічні властивості.

У багатьох випадках, наприклад, у випадку RSA, великі прості числа є ключовими параметрами.

Відповідні обчислювальні процедури включають в себе алгоритми, що реалізують етап перевірки чисел на простоту.

В літературі і криптографічній практиці подібні алгоритми носять отримали назву тестів.

Детерміновані тести дозволяють довести, що число, яке тестується, – просте. Практично застосовувані детерміновані тести здатні дати позитивну відповідь не для кожного простого числа, оскільки використовують лише достатні умови простоти.

Детерміновані тести більш корисні, коли необхідно побудувати випадкове велике просте число, а не перевірити простоту, скажімо, деякого єдиного числа.

Детермінований тест використовується, наприклад, в процедурах обчислення несекретних параметрів цифрового підпису типу Ель-Гамаля, встановлених ГОСТ 34.310.

Для реалізації дослідження розглянуті детерміновані тести при побудові асиметричних криптосистем створення ЕЦП, теорема Демітко, тест на основі малої теореми Ферма, основні властивості псевдопростих чисел, властивості чисел Кармайкла, тест Соловея-Штрассена і Ейлерові псевдопрості числа, приклад побудови сильного простого числа, метод Гордона побудови сильних простих чисел.

3.11 Перелік джерел посилань до розділу 3

1. Криптологія у прикладах, тестах і задачах: навч. посібник. / Т. В. Бабенко, Г. М. Гулак, С.О. Сушко, Л. Я. Фомичова.– Д.: Національний гірничий університет, 2013. – 318 с.

2. Теорема Демітко. [Електронний ресурс]. – Режим доступу: http://www.immsp.kiev.ua/postgraduate/Biblioteka_trudy/Osnovy_KZI_Gulak.G.M_Muchatchev.V.A_2011.pdf (дата звернення 24.10.2020).

3. Тест на основі малої теореми Ферма. [Електронний ресурс]. – Режим доступу: https://ozlib.com/869442/informatika/test_prostoty_osnove_maloy_teoremy_ferma (дата звернення 24.10.2020).

4. Псевдопростое число. [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Псевдопростое_число (дата звернення 24.10.2020).

5. Функция Кармайкла. Електронний ресурс]. – Режим доступу: http://poivs.tsput.ru/ru/Math/NumberTheory/General/SeverabilityAndResidues/CarmichaelFunction (дата звернення 24.10.2020).

6. Алгоритм Соловея-Штрассена. Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/127544/ (дата звернення 24.10.2020).

7. Тест Миллера – Рабина – проверка числа на простоту. Електронний ресурс]. – Режим доступу: https://vscode.ru/prog-lessons/test-millera-rabina-na-prostotu-chisla.html (дата звернення 24.10.2020).

8. Метод Гордона побудови сильних простих чисел. Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5367441/page:8/ (дата звернення 24.10.2020).

9. Числа Эйлера. Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Числа_Ейлера (дата звернення 24.10.2020).

РОЗДІЛ 4

ТЕСТУВАННЯ ПРОГРАМНИХ КОМПЛЕКСІВ СТВОРЕННЯ ЕЦП

4.1 Тестування програмних засобів створення ЕЦП та шифрування

Для того щоб визнати електронний документ рівним з юридичної точки зору паперовому з підписом і печаткою, необхідний електронний підпис, а безпечна відправка чутливих даних повинна бути забезпечена за допомогою їх шифрування. Але незважаючи на те, що держава всіляко пропагує перехід до електронної взаємодії, нормативно-правова база, що регулює напрямок PKI, розвинена слабо, в ній відсутні певні визначені принципи побудови інфраструктури відкритих ключів.

Один з основних нормативно-правових «прогалин» в області PKI - відсутність чіткого порядку перевірки дійсного статусу сертифіката ключа перевірки електронного підпису. Це дає розробникам засобів, що реалізують технології електронного підпису та шифрування, свободу в реалізації даної перевірки.

В умовах, що склалися раціонально орієнтуватися на кращі світові практики в області PKI - наприклад, тести, розроблені Національним інститутом по стандартам і технологіям США (National Institute of Standards and Technology, далі - NIST). Вони допомагають визначити коректність реалізованої засобом електронного підпису та шифрування даних перевірки статусу сертифіката ключа перевірки електронного підпису.

Опис тестів можна отримати на сайті Національного інституту по стандартам та технологіям США [1].

У документі описано 224 перевірки, для реалізації кожної з яких створюється сертифікат або набір сертифікатів ключів перевірки електронного підпису, що встановлюються в сховище комп'ютера. Далі за допомогою засобу електронного підпису та шифрування виконується аналіз того, наскільки коректно відображається статус сертифіката в інтерфейсі програмного засобу.

4.2 Загальна схема створення ЕЦП

Незалежно від застосовуваного алгоритму загальна схема ЕЦП в системі асиметричного шифрування може бути представлена наступним чином [2]. На першому кроці обчислюється хеш-функція h від переданого документа (Повідомлення) M і створюється дайджест документа $m = h(M)$. За допомогою секретного ключа відправника A

k_A і алгоритму формування ЕЦП *E* створюється зашифрований дайджест повідомлення M C (m). У пакет для одержувача *B* включаються: повідомлення M, ЕЦП C (m) і відкритий ключ відправника K_A, пакет передається одержувачу *B* по відкритому каналу зв'язку (відзначимо, що відкритий ключ K_A може не бути передано по каналу зв'язку, а публікуватися будь-яким іншим чином, наприклад, розміщуватися на сайті) (рис. 4.1).

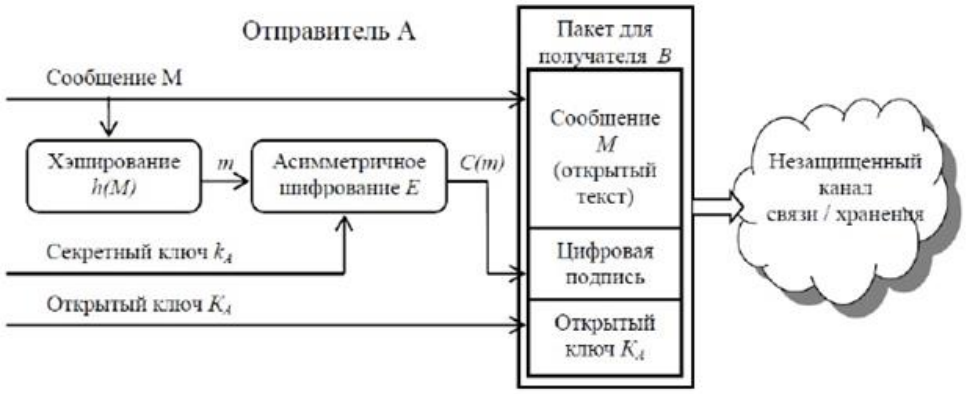


Рисунок 4.1 – Загальна схема створення ЕЦП

На етапі верифікації підпису одержувач B обчислює хеш-функцію h (M) і отримує дайджест m', розшифровує ЕЦП алгоритмом дешифрування D, отримує при цьому дайджест m'. Далі проводиться порівняння двох хеш-функцій m' і m. Їх збіг гарантує одночасно справжність вмісту документа і його авторства, в разі розбіжності підпис відкидається (рис.2).

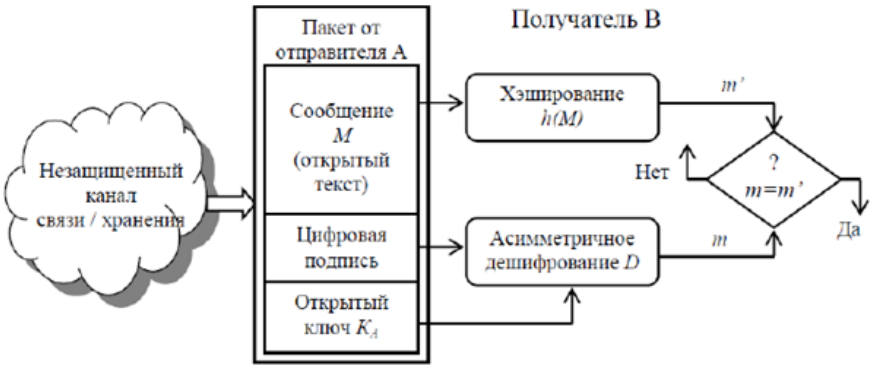


Рисунок 4.2 – Етап верифікації

4.3 Практичне застосування алгоритмів створення ЕЦП

Розглянемо найбіль вживані сучасні алгоритми ЕЦП, що застосовуються на практиці. Всі вони і є асиметричними, тобто використовують відкритий ключ. За типом застосовуваної односторонньої функції з лазівкою (One-way function with trapdoor) вони діляться на системи засновані на:

– факторизації добутку двох великих простих чисел;

– обчисленні дискретного логарифма в кінцевому полі;

– задачі дискретного логарифмування на еліптичних кривих у кінцевому полі.

Алгоритм RSA (абревіатура від прізвищ творців Rivest, Shamir і Adleman) - перший практичний криптографічний алгоритм з відкритим ключем, грунтується на обчислювальної складності задачі факторизації великих цілих чисел. Був створений в Массачусетському технологічному інституті (MIT) в 1977 р. Захищений патентом США 4405829, виданим 20 вересня 1983 року і діє по теперішній час.

У 1982 році Ривест, Шамір і Адлеман організували компанію RSA Data Security, яка є єдиним власником і розповсюджувачем алгоритму.

На сьогоднішній день RSA є фактично неофіційними світовим комерційним стандартом ЕЦП, це самий застосовуваний в світі алгоритм підпису, за деякими експертними оцінками їм підписується до 90% всіх документів. Однак головним недоліком RSA є його закритий код і необхідність придбання ліцензії.

Алгоритм DSA (Digital Signature Algorithm) розроблений Національним інститутом стандартів і технологій США (ність) в серпні 1991 р. і захищений патентом США 5 231 668, проте ність зробив цей патент доступним для використання без ліцензійних відрахувань. Оскільки алгоритм став вільним для використання, його можна вільно реалізовувати програмним, апаратним або будь-яким іншим чином. Заснований на складності завдання дискретного логарифмування.

Алгоритм разом з криптографічного хеш-функцією SHA-1 є частиною стандарту США DSS (Digital Signature Standard), вперше опублікованого в 1994 р. Решта розглянуті алгоритми засновані на еліптичній криптографії - розділі криптографії, який вивчає асиметричні криптосистеми, засновані на еліптичних кривих над кінцевими полями. Основна перевага еліптичної криптографії полягає в тому, що на сьогоднішній день невідомо існування субекспоненціальних алгоритмів розв'язання задачі дискретного логарифмування.

Алгоритм ECDSA (Elliptic Curve Digital Signature Algorithm) - алгоритм з відкритим ключем для створення цифрового підпису, аналогічний, за своєю будовою, DSA, але певний, на відміну від нього, не над полем цілих чисел, а в групі точок еліптичної кривої. ECDSA є

дуже привабливим алгоритмом для реалізації ЕЦП. Найважливішою перевагою ECDSA є можливість його роботи на значно менших кінцевих полях. Як, і взагалі в криптографії на еліптичних кривих, передбачається, що бітовий розмір відкритого ключа, який буде необхідний для ECDSA, дорівнює подвійному розміру секретного ключа в бітах.

Для порівняння, при забезпеченні рівня безпеки в 80 біт (тобто коли атакуючому повним перебором необхідно розглянути приблизно 2^{80} версій підпису для знаходження секретного ключа), розмір відкритого ключа DSA дорівнює, принаймні, 1024 біт, тоді як відкритого ключа ECDSA - 160 біт. З іншого боку розмір підписи однаковий і для DSA, і для ECDSA: 4t біт, де t - рівень безпеки, виміряний в бітах, тобто - приблизно 320 біт для забезпечення рівня безпеки у 80 біт.

Будь-яка криптосистема на еліптичній кривій забезпечує ту ж криптографічну стійкість, що і система, заснована на дискретному логарифмуванні при істотно меншій довжині ключа.

Алгоритм ГОСТ Р34.10-2012 – російський стандарт, що описує алгоритми формування та перевірки електронного цифрового підпису. Прийнятий і введений в дію у 2012 р. на основі алгоритму, який був прийнятий ще у 1994 р. [3]

Алгоритм ДСТУ 4145-2002 (повна назва: «ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, заснована на еліптичних кривих. Формування та перевірка ») - український стандарт, описує алгоритми формування та перевірки електронного цифрового підпису [4]. Прийнятий і введений в дію в 2002 р Незважаючи на досить довгий існування ДСТУ 4145-2002 в якості державного стандарту ЕЦП, він залишається декларованим, але мало придатним, незважаючи на те, що його основні криптографічні примітиви реалізовані в відкритих Java-бібліотеках OpenSSL і Bouncy Castle [5].

Існує дві причини цього:

– дуже складний виклад тексту самого стандарту; хоча офіційне видання містить всього 39 сторінок, читання документа і його розуміння займає тривалий час навіть у достатньо підготовлених професійних програмістів;

– практично повна відсутність можливостей розпаралелювання при програмній реалізації алгоритму.

Результати попереднього аналізу всіх перерахованих алгоритмів зведені в таблицю 4.1.

Таблиця 4.1 – Основні характеристики алгоритмів ЕЦП

Алгоритм	Хеш-функція	Рекомендований розмір відкритого ключа, біт	Рекомендований розмір закритого ключа	Рік створення, країна
DSA	SHA-1 або SHA-2	1024-3072	160-256	1994, США
ECDSA	SHA-1 або SHA-2	112-320	80-512	1999, США
ГОСТ Р34.10-2012	ГОСТ Р34.112012	80-320	256-512	2012, РФ
ДСТУ 4145-2002	ГОСТ 34311.95	162-768	256-1024	2002, Україна

Не зупиняючись детально на огляді застосовуваних в ЕЦП криптографічних хеш-функціях, відмітимо що всі вони засновані на алгоритмах блокового шифрування.

В даному дослідженні ЕЦП були застосовані хеш-функції MD-2 MD-5 SHA-1 SHA-2 (з розмірами блоку 256, 384, 512 біт), також ГОСТ 34311.95 і ГОСТ Р34.112012, специфіковані відповідними стандартами ЕЦП [6].

4.4 Програмний засіб для моделювання і порівняльного аналізу систем ЕЦП

Система спроектована відповідно до основних вимог компонентного програмування. Кожний модуль системи відповідає за реалізацію чіткого та мінімального переліку функцій та взаємодії з іншими компонентами за допомогою гнучкого інтерфейсу.

Для роботи web-інтерфейсу необхідно мати сучасний web-браузер та стабільний доступ до мережі Інтернет.

Для запуску системи на сервері необхідно принаймні 2 гігабайти оперативної пам’яті та Java Development Kit версії 8 або вище.

Шар представлення, реалізований у вигляді web-інтерфейсу, сполучається контролером із шаром сервісу, в якому можна відокремити самостійні елементи, відповідальні за збереження інформації про користувачів та безпосередньо створення і перевірку цифрового підпису.

Для порівняльного аналізу систем ЕЦП шляхом моделювання розроблено універсальний програмний засіб з наступними функціями:

– блок хешування;

– блок генерації відкритого ключа;

– блок генерації секретного ключа;

– блок формування ЕЦП;

– блок верифікації ЕЦП.

Принципову архітектуру системи зображено на рисунку 4.3.

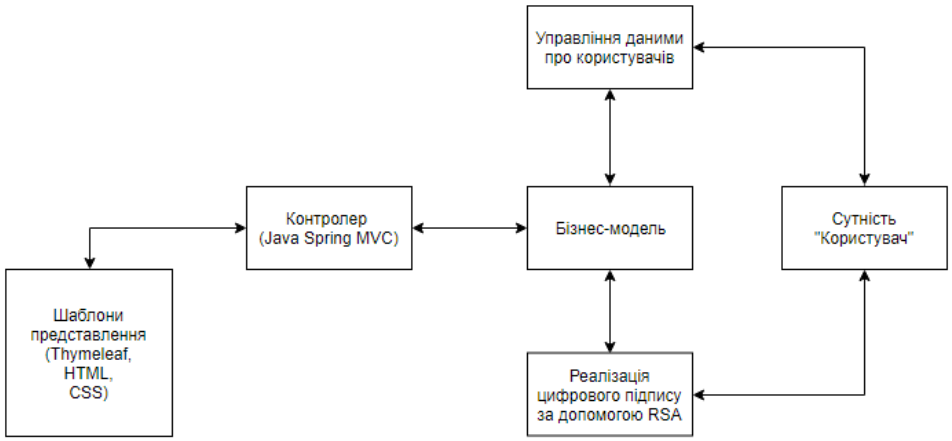


Рисунок 4.3 — Загальна архітектура створеної системи

Програма розроблена на мові Java з застосуванням бібліотеки криптографічних примітивів Open SSL. Open SSL (secure socket layer - система безпечних сокетів) - криптографічний пакет з відкритим вихідним кодом, що надає багаті засоби для профілювання програмних фрагментів за витратами процесорного часу [7].

Деякі модулі, наприклад опис структури даних про користувача, використовуються одночасно у роботі більш ніж одного іншого модуля. Можливість повторного використання характеризує високу відповідність системи стандартам об’єктно-орієнтованого програмування.

Виконання певних дій передбачає взаємодію з файловою системою користувача.

Головною складовою системи, де реалізовані всі сервіси, які вона може надавати користувачам, є її бізнес-модель.

Дана частина системи реалізує функції:

— підпису тексту;

— перевірки правильності підпису для конкретного тексту;

— зчитування даних про користувачів зі сховища даних;

— оновлення даних про користувачів у сховищі даних.

Дані про користувача зберігаються у класі, що містить такі поля даних:

— ім’я користувача;

— пароль персонального кабінету;

— пара унікальних RSA ключів;

— об’єкт цифрового підпису, що отримує початкові значення своїх параметрів при створенні кабінету.

Діаграма прецедентів, що зображена на рисунку 4.4, описує набір дій, які створюють цілісний та повний функціональний сценарій роботи користувача з системою.

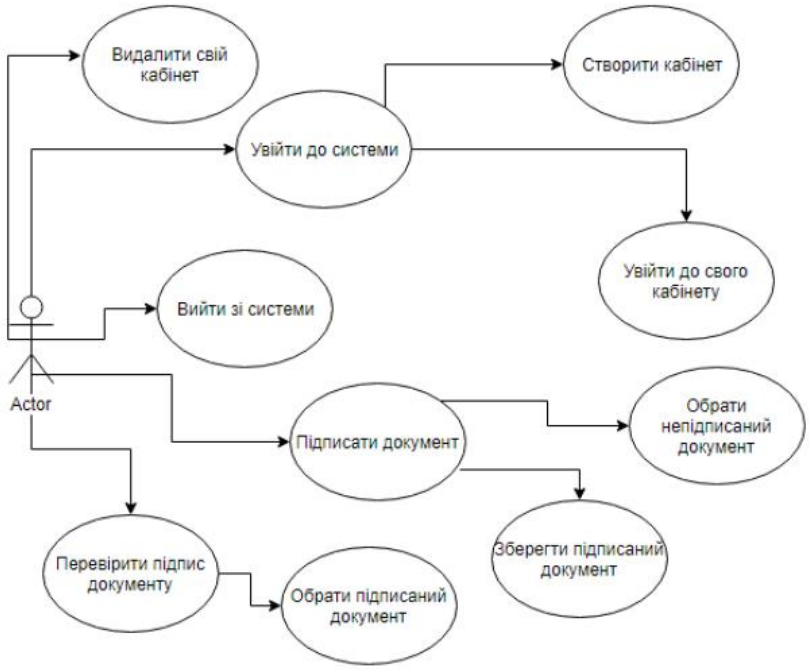


Рисунок 4.4 — Діаграма прецедентів системи

Задачею модулю взаємодії зі сховищем даних є взаємна конвертація між вище описаною структурою даних та фізичним форматом, у якому вони зберігаються у певних файлах.

Аналізуючи поставлену задачу, вирішено розроблювати програмне забезпечення у вигляді веб-додатку. Це оптимальне рішення, тому що мета роботи передбачає застосування програми користувачами, які знаходяться віддалено один від одного. Клієнт веб-застосунків не операційна система, а встановлений браузер, що дозволяє зробити систему універсальнішою та завідома сумісною з більшою кількістю пристроїв.

4.5 Засоби розробки

Логіка додатку та внутрішні обчислювальні методи реалізовані за допомою мови програмування JAVA 8 версії.

Середовищем розробки обрано IntelliJ IDEA.

Для створення користувацького інтерфейсу використано мову розмітки HTML та мову стилів CSS.

Для збірки проекту використано фреймворк Maven.

Для зручності написання та побудови структури проекту використано фреймворк Spring.

Використані зовнішні бібліотеки, наприклад BitInteger для множення великих чисел без колізії.

В якості сховища даних обрано файл json формату, дані в якому видозмінені за допомогою методів серіалізації/десеріалізації.

4.5.1 Мова програмування Java

Java є мовою програмування високого рівня та обчислювальною платформою, розробленою компанією Sun Microsystems у 1995 році. З тих пір мова регулярно поповнюється новими версіями.

Виходячи з переваг Java, вона отримала широку популярність. Побудовано декілька конфігурацій для різних типів платформ, включаючи Java SE для Macintosh, Windows і UNIX, Java ME для мобільних додатків і Java EE для корпоративних додатків.

Завдяки зростаючому значенню веб-додатків і мобільних додатків, Java сьогодні є основою для більшості мережевих програм і вважається корисною для серверної розробки, роботою з веб-вмістом, створення корпоративного програмного забезпечення, ігор і мобільних додатків.

Переваги Java:

- Java пропонує більш високу крос-функціональність і портативність, оскільки програми, написані на одній платформі, можуть працювати на настільних комп'ютерах, мобільних телефонах, вбудованих системах;

- проста, об'єктно-орієнтована, розподілена;

- підтримує багатопоточність;

- Java є безкоштовною, і пропонує мультимедійну та мережеву підтримку;

- Java - це зріла мова, тому вона більш стабільна і передбачувана. Бібліотека класів Java дає можливість розробки на різних платформах;

- дуже популярна на корпоративному, вбудованому та мережевому рівнях, Java має велику активну спільноту користувачів і доступну підтримку;

- підвищення мовної різноманітності, про що свідчить сумісність Java з Scala, Groovy, JRuby і Clojure;

- відносно бездоганна сумісність з однієї версії до іншої.

В кожній версії Java є свої особливості. Для кодування системи було обрано версію 8 цієї мови, оскільки її можна вважати найбільш стабільною та поширеною, без експериментальних доповнень, які не пройшли перевірку часом та достатньою кількістю реальних проектів.

4.5.2 Фреймворк Maven

Maven - це інструмент управління проектами, який надає розробникам готову структуру життєвого циклу [7]. Розробник може автоматизувати побудову інфраструктури проекту в короткий час, оскільки Maven використовує стандартну розкладку каталогів і побудову життєвого циклу за замовчуванням. Maven може налаштувати спосіб роботи відповідно до стандартів за дуже короткий час. Оскільки більшість установок проекту є простими і довговічними, Maven спрощує процес розробки під час створення звітів, перевірок, побудови структури проекту та автоматизованого тестування. Maven надає розробникам ряд переваг:

- управління та впровадження повного циклу проекту. Компіляції коду, тестування модулів, генерація збірки (файли jar та war);

- легке керування зовнішніми залежностями проекту на основі Java і зберігання проекту Java невеликим.

- багаторівневе управління проектами, тобто обмін функціональними можливостями одного проекту з іншими численними проектами з використанням управління залежностями.

- Краще звітування про помилки та цілісність - Maven покращила звітність про помилки, і вона надає вам посилання на вікі-сторінку Maven, де ви отримаєте повний опис помилки.

Підводячи підсумок, Maven спрощує і стандартизує процес створення проекту. Він обробляє компіляцію, розповсюдження, документацію, командну співпрацю та інші завдання. Maven підвищує можливості повторного використання і піклується про більшість завдань, пов'язаних з будовою структури проекту.

4.5.3 Фреймворк Spring

Spring - найпопулярніша платформа розробки додатків для Java. Мільйони розробників у всьому світі використовують Spring Framework для створення високопродуктивного та повторно використовуваного коду, що легко тестується [8].

Spring framework - відкрита платформа Java. Він був написаний Родом Джонсоном і вперше був випущений під ліцензією Apache 2.0 у червні 2003 року. Основні особливості Spring Framework можна використовувати при розробці будь-якого Java-додатка, для створення веб-додатків поверх платформи Java EE. Цільові функції Spring сприяють хорошій практиці програмування.

Переваги використання Spring Framework:

- Spring дозволяє розробникам розробляти програми з використанням POJO. Перевагою використання тільки POJO є те, що вам не потрібен контейнер EJB, такий як сервер додатків;

- Spring організований модульно. Незважаючи на те, що кількість пакетів і класів є істотною, вам потрібен лише неширокий їх спектр для кожної задачі.

- Spring не винаходить колесо, замість цього він дійсно використовує деякі з існуючих технологій, таких як кілька ORM фреймворків, JEE, Quartz і JDK таймери, а також інші технології перегляду та відображення.

Веб-фреймворк Spring - це добре продуманий веб-фреймворк, базований на MVC, який надає чудову альтернативу веб-фреймворкам, таким як Struts або інші надмірно складні або менш популярні веб-фреймворки.

Технологія, з якою найбільш ототожнюється Spring, - це Dependency Injection (ін'єкції залежності) від інверсії контролю. Інверсія контролю є загальною концепцією, і вона може бути виражена багатьма різними способами.

ін'єкція залежностей може відбуватися шляхом передачі параметрів конструктору або POST запиту з використанням методів сеттера.

4.5.4 Thymeleaf шаблонізатор

Thymeleaf - бібліотека Java [9]. Це шаблонний движок XML/XHTML/HTML5, здатний застосувати набір перетворень до шаблонів для відображення даних та тексту, створених вашими програмами. Він краще підходить для обслуговування XHTML/HTML5 у веб-додатках, але він може обробляти будь-який файл XML, як в Інтернеті так і в окремих програмах.

Основна мета Thymeleaf - забезпечити елегантний і добре сформований спосіб створення шаблонів. Щоб досягти цього, він базується на тегах XML і атрибутах, які визначають виконання попередньо визначеної логіки на DOM (Document Object Model), замість того, щоб явно писати цю логіку як код всередині шаблону.

Його архітектура дозволяє здійснювати швидку обробку шаблонів, спираючись на інтелектуальне кешування проаналізованих файлів, щоб використовувати найменш можливу кількість операцій введення / виводу під час виконання. Важливо, що Thymeleaf було розроблено з самого початку з урахуванням XML і веб-стандартів, що дозволяє створювати шаблони, які легко перевіряються та редагуються, якщо це необхідно для вас.

Головною метою Thymeleaf є створення елементарних природних шаблонів у вашому робочому процесі розробки - HTML, який можна правильно відобразити в браузерах, а також стати статичними прототипами, що дозволяє інтегрувати структурні елементи програми в новий вигляд.

Eclipse, IntelliJ IDEA, Spring, Play, навіть найсучасніші API для моделей Model View Controller nf Java EE 8. Thymeleaf підтримує всі популярні інструменти. Thymeleaf ідеально підходить для сучасної веб-розробки HTML5 в рамках Java проектів. Шаблони HTML, написані в Thymeleaf, виглядають і працюють як HTML, дозволяючи реальним шаблонам, які виконуються у вашому додатку, працювати як корисні артефакти дизайну.

4.5.5 Розмітка HTML 5 та стилі CSS 3

HTML (HyperText Markup Language) - мова розмітки гіпертексту, призначена для створення Web-сторінок. Під гіпертекстом в цьому випадку розуміється текст, пов'язаний з іншими текстами покажчиками-посиланнями [10].

HTML являє собою досить простий набір кодів, які описують структуру документа. HTML дозволяє виділити в тексті окремі логічні частини (заголовки, абзаци, списки і т.д.), помістити на Web-сторінку підготовлену фотографію або картинку, організувати на сторінці посилання для зв'язку з іншими документами. Будь-який HTML документ складається з набору елементів, початок і кінець кожного елемента позначається спеціальними позначками - тегами. Між тегами ви побачите уміст елемента - дані або текст (наприклад елемент заголовка: <title> Hello, world! </ Title>, де <title> - відкриває тег, </ title> - закриває). Крім даних, елемент може мати атрибути, що визначають властивості елемента (наприклад спосіб вирівнювання тексту, колір шрифту, розмір картинки).

Мова HTML5 робить акцент на спрощення розмітки, необхідної для створення відповідних стандартів сторінок і об'єднання всього необхідного з CSS і Java кодом, а також файлів зображень.

HTML можна назвати основною мовою Всесвітньої павутини. Більшість веб-сторінок, розміщених в Інтернеті, написані в якійсь з варіацій HTML. За допомогою нього розробники визначають те, як мультимедіа, текст або гіперпосилання відображатимуться серед іншого контенту в браузері. Починаючи від елементів, які встановлюють зв'язку з вашим документом до елементів які роблять ці документи інтерактивними (наприклад форми) - все це є складовими частинами HTML. CSS - це фактично мова стилів, який визначає відображення HTML-документів. CSS працює зі шрифтами, з символами і фоном, полями, рядками, з висотою і з шириною елементів відображення, з фоновими зображеннями, з позиціонуванням елементів і тому подібне. Якщо HTML необхідний для структурування змісту сторінки, то CSS необхідний для того, щоб форматувати цю структуру.

Використання CSS полегшує створення якісних сайтів, дозволяючи задати стилі окремих елементів сторінок сайту в особливих css-файлах, щоб в подальшому бути впевненим в тому, що всі сторінки сайту будуть витримані в єдиному стилі.

Для форматування сторінок користувацького інтерфейсу використана 3 версія мови CSS, яка має свої переваги:

- використання таблиці стилів. Задовго до розробки концепції каскадних таблиць стилів CSS3 використовувалася HTML-розмітка. Але з введенням каскадних таблиць все це стало можливим задавати в окремій таблиці стилів, в результаті чого користувачі отримали простий та зручний інструмент. З цим пов'язаний ще один плюс CSS3 - стало простіше вносити зміни: можна змінювати окремі модулі, які інтегруються із загальною системою.

- диференціація і ізоляція. Диференціація забезпечує більш ефективну і зручну форму. Модульний підхід допомагає розвивати і підтримувати системи, даючи велику гнучкість.

- макет декількох колонок. Модуль на основі декількох колонок (Multi-Column Module) - важлива функція CSS3, що дозволяє помістити текст в декілька стовпців.

- гнучкість у використанні. Концепція каскадних таблиць стилів дозволяє приєднати інформацію CSS-стилю у вигляді окремого документа або у вигляді вкладення усередині HTML-документа. Також можна імпортувати кілька таблиць стилів в будь-якому місці.

4.6 Середовище розробки

Без сумніву, найпопулярнішим середовищем для написання коду на мові програмування Java є IntelliJ IDEA від Jet Brains.

Використано новітнє та поширене середовище, щоб уникнути часу на вивчення принципів роботи з технологіями, які рідко використовуються для розробки подібних проектів, але отримати доступ до достатньо великого та ефективного набору засобів редагування та налаштування Java-проектів.

IntelliJ IDEA - це, перш за все, середовище розробки для Java, включаючи Java 8. З цією мовою вона дружить найбільше, відмінно його розуміє і допомагає в написанні розробнику. Але це не означає, що все закінчується на Java і власною мовою Kotlin. Не менш важливими є їх розробки таких передових технологій, як Groovy, Scala і інших. Вони поєднують в собі можливості динозаврів, як Java разом з функціоналом Ruby, Smalltalk і інших. Однією з сильних сторін в JetBrains вважають підтримку широкого кола технологій.

IntelliJ IDEA являє собою високотехнологічний комплекс тісно інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторинга коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant / JUnit і системами управління версіями, унікальний інструмент оптимізації та перевірки коду Code Inspection, а також інноваційний візуальний конструктор графічних інтерфейсів.

Модуль електронного підпису також взаємодіє з даною структурою, коли використовує ключі шифрування конкретного користувача з метою створення чи перевірки електронного підпису.

На рис. 4.5 приведена основна екранна форма програми.

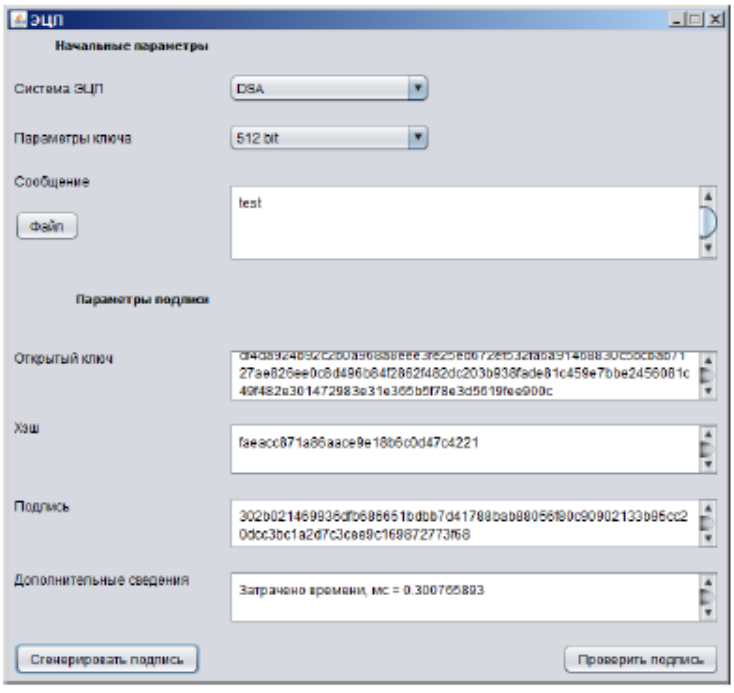


Рисунок 4.5 – Основна екранна форма програми

Моделювання ЕЦП в цілому і хеш-функцій проводилося на апаратно-програмних конфігураціях двох типів:

1. Конфігурація А:

– CPU: Pentium 987 (ядро Sandy Bridge) 1.5 ГГц (2 МБ L1 cache);

– RAM: 4 ГБ DDR3 1300 МГц;

– ОС: Win7.

2. Конфігурація Б:

– CPU: Core i5 3240T (ядро Ivy Bridge) 2.9 ГГц (3 МБ L1 cache);

– RAM: 8 ГБ DDR3 1600 МГц;

– ОС: Win7.

Конфігурація А моделює типовий сучасний офісний комп'ютер в організаціях і компаніях, які використовують ЕЦП, а конфігурація Б – типовий комп'ютер, який застосовується в підрозділах комп'ютерної безпеки банків і інших фінансових установ та верифікує ЕЦП.

4.7 Результати досліджень

В процесі моделювання досліджена продуктивність (швидкість роботи) хеш-функцій. Усереднені по базі файлів документів різних форматів і розмірів обсягом $3 \cdot 10^4$ файлів. Результати дослідження швидкодії хеш-функцій наведені в таблиці 2 і на рис. 4. 6. Показники швидкості оцінені програмними засобами Java і Open SSL.

За даними таблиці 4.2 та рисунку 4.6 слідує, що хешування за допомогою SHA-2 з довжиною блоку 512 біт є найбільш продуктивною обчислювальною процедурою. З іншого боку, алгоритм SHA-2 є досить сучасною і перспективною функцією хешування, що гарантує її криптографічну стійкість.

Таблиця 4.2 – Результати дослідження швидкодії роботи хеш-функцій

Функція хешування	Кількість раундів	Мова реалізації	Швидкість роботи на конфігурації А, Мбіт/с	Швидкість роботи на конфігурації Б, Мбіт/с
SHA-1	80	Java	206	344
SHA-2 (256)	64	Java	81	135
SHA-2 (512)	64	Java	41	68
ГОСТ 34311.95	256	Java	4928	83
ГОСТ 34.112012	256	Java	2458	46

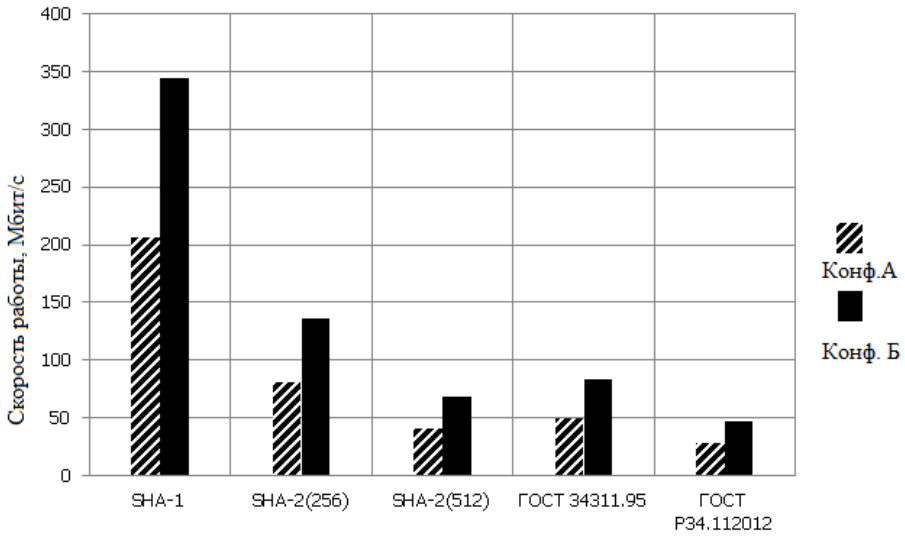


Рисунок 4.6 – Порівняльний аналіз швидкодії роботи хеш-функцій

При дослідженні алгоритмів ЕЦП був проведений попередній аналіз систем ЕЦП, наведених вище в таблиці 1. Він дозволив вивести м'яку рейтингову оцінку, згідно з якою програмна реалізація алгоритмів ГОСТ Р34.10-2012 і ДСТУ 4145-2002 істотно складніше алгоритмів DSA і ECDSA, тому при необхідних довжинах ключів ГОСТ Р 34.10-2012 і ДСТУ 4145-2002 практично не витримують конкуренції з алгоритмами DSA і ECDSA по швидкодії. Нижче наведені результати аналізу швидкодії тільки для найбільш швидких систем ЕЦП DSA і ECDSA.

Оцінка швидкодії систем ЕЦП DSA і ECDSA, в залежності від довжини ключа з розбивкою по етапах, представлені на рис. 4.7 - 4.8 для обох апаратних конфігурацій.

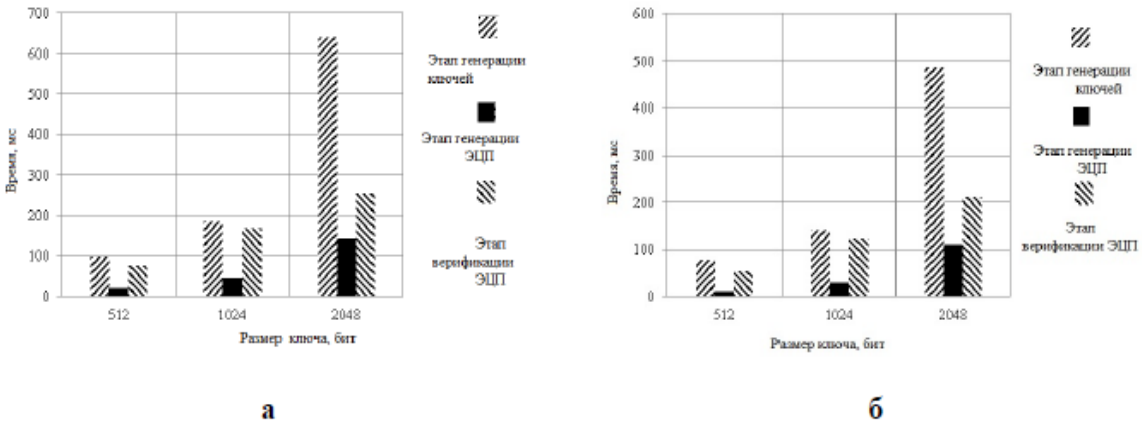


Рисунок 4.7 – Часовий аналіз етапів DSA в залежності від розміру ключа:
а) конфігурація А, б) конфігурація Б

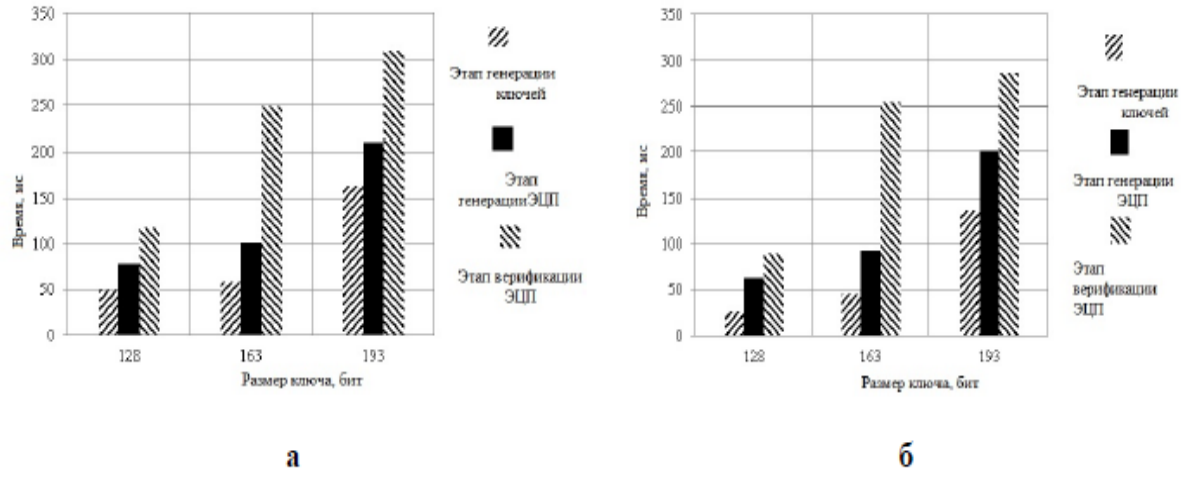


Рисунок 4.8 – Часовий аналіз етапів ECDSA в залежності від розміру ключа:
а) конфігурація А, б) конфігурація Б

Аналіз графічних залежностей дозволяє зробити наступні висновки .

Підпис на еліптичних кривих вимагає істотно більшого часу на верифікацію, ніж на формування ключів. Для користувальницької конфігурації А цілком прийнятно з точки зору швидкодії для системи DSA рекомендувати довжину ключа 1024 біта, а для системи ECDSA - довжину ключа 163 біта. Для професійної конфігурації Б ці рекомендації зберігаються.

Загальні рекомендації, отримані в результаті дослідження, сформулюємо таким чином. Найбільш ефективними по сформульованим критеріям зручності програмної реалізації і швидкодії є системи ЕЦП DSA і ECDSA з застосуванням хеш-функції SHA-2 довжиною блоку 256 або 512 біт. при порівнянні систем DSA і ECDSA останню, безумовно, слід вважати більш перспективною, оскільки більшість діючих стандартів ЕЦП орієнтовані на еліптичну криптографію.

4.8 Методологія тестування засобів електронного підпису та шифрування документів

Щоб оцінити роботу представлених програмних засобів, що реалізують технології електронного підпису та шифрування документів, на основі тестів NIST виконано порівняльне тестування коректності їх роботи. Для порівняння були відібрані такі програмні продукти:

1. Litoria Desktop (збірка 1.0.44).
2. КриптоАРМ (збірка 5.4.1.37).

3. Admin-PKI (збірка 5.1.1.1)

4. КАРМА (збірка 56.0.80).

5. КриптоНУЦ (збірка 1.12.2)

6. КріптЕК-Д (демоверсія 1.1.3.42)

7. File- PRO (збірка 2.4.0.15).

8. VipNet CryptoFile (збірка 4.0.1.43722)

Слід відзначити, що програмні комплекси КріптЕК-Д, File-PRO і VipNet CryptoFile не дають можливості переглянути статус сертифіката через власний інтерфейс. Відповідно, зробити висновки щодо коректності їх функціонування не представляється можливим (рисунки 4.9-4.11). Тому з тестування вони виключені.

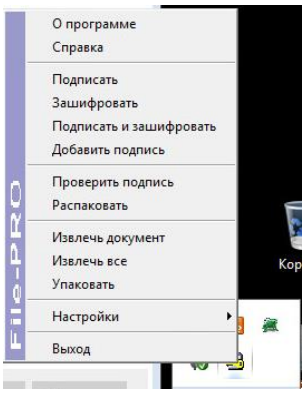


Рисунок 4.9 – Реалізація функцій формування електронного підпису та шифрування даних за допомогою File-PRO (збірка 2.4.0.15)

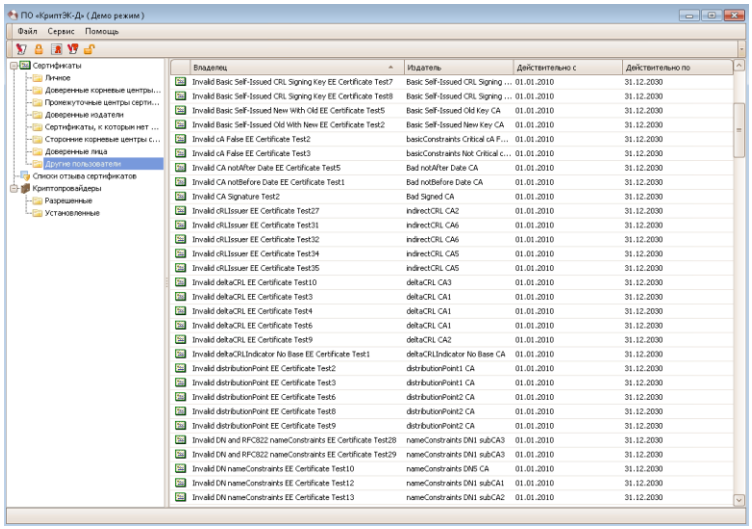


Рисунок 4.10 – Відображення сертифікатів, встановлених в сховище комп'ютера в інтерфейсі КріптЕК-Д (демоверсія 1.1.3.42)

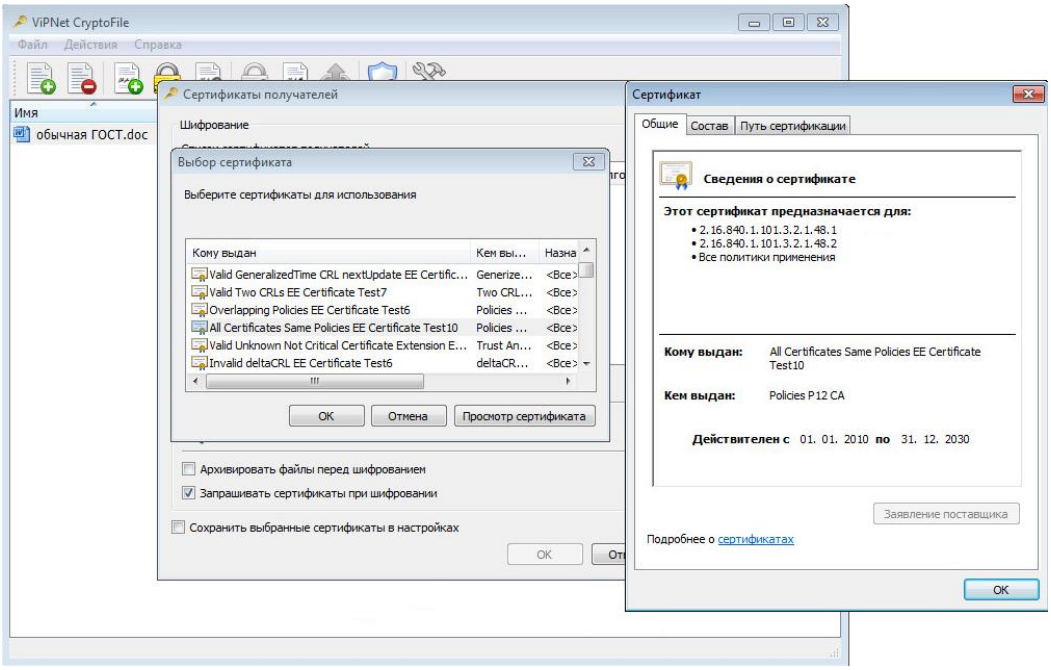


Рисунок 4.11 – Відображення сертифікатів, встановлених в сховище комп'ютера в інтерфейсі VipNet CryptoFile (збірка 4.0.1.43722)

У підсумковій таблиці 4.3 приведені результати проходження тестів по NIST зазначених програм.

Таблиця 4.3 – Підсумкові результати проходження тестів по NIST

Litoria Desktop	КриптоАРМ	Admin-PKI	КАРМА	КриптоНУЦ
Пройдено тестів				
224	160	173	166	168
Не пройдено тестів (причина – статус сертифікату ЕЦП визначений невірно)				
0	10	18	13	13
Не пройдено тестів (причина – результат невірний)				
0	34	13	25	23
Число нереалізований тестів				
0	20	20	20	20

З більш розширеним звітом тестування зазначених програмних комплексів можна ознайомитись за посиланням [11].

Виходячи з результатів тестування беззаперечним лідером є програма Litoria Desktop, яка пройшла всі тести по NIST та може бути рекомендована до використання при створенні ЕЦП.

4.9 Висновки до розділу 4

У четвертому розділі магістерської роботи проведено дослідження сучасних систем ЕЦП шляхом їх програмного моделювання.

За критеріями зручності програмної реалізації і швидкодії це дозволило визначити , що вони в найбільшій мірі задовольняють сформульованим критеріям і можуть бути рекомендовані для практичного застосування.

В результаті дослідження вироблені практичні рекомендації по довжині блоків хеш-функції і довжині ключа алгоритмів ЕЦП.

Таким чином, дослідження на основі тестів, запропонованих NIST, показало, наскільки недосконалі засоби електронного підпису та шифрування документів.

З цього випливає, що поряд з перевірками коректності інтеграції з криптопровайдером, проведеними для рішень, що забезпечують реалізацію технологій електронного підпису та шифрування (що в свою чергу покладено на плечі регуляторів), величезне значення має і тестування правильності реалізації функцій, для яких це рішення створювалося.

Адже помилка в правильності перевірки статусу сертифіката може спричинити фінансовий і репутаційний збиток як для самого розробника рішення, так і для користувачів, які експлуатують це рішення.

4.10 Перелік джерел посилань до розділу 4

1. Public Key Interoperability. Test Suite (PKITS). Certification Path Validation. [Электронный ресурс]. – Режим доступу: http://csrc.nist.gov/groups/ST/crypto_apps_infra/documents/PKITS.pdf (дата звернення 21.11.2020).

2. Використання механізму електронного цифрового підпису. [Електронний ресурс]. – Режим доступу: https://www.znanius.com/3852.html 9/ (дата звернення 24.10.2020).

3. ГОСТ Р 34.10-2012. [Електронний ресурс]. – Режим доступу: http://docs.cntd.ru/document/gost-r-34-10-2012 (дата звернення 24.10.2020).

4. Алгоритм ДСТУ 4145-2002. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/ДСТУ_4145-2002 (дата звернення 24.10.2020).

5. Использование библиотеки OpenSSL в проектах на C++. [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/277935/ (дата звернення 24.10.2020).

6. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. [Електронний ресурс]. – Режим доступу: http://docs.cntd.ru/document/gost-r-34-11-2012 (дата звернення 24.10.2020).

7. Краткое знакомство с Maven. [Електронний ресурс]. – Режим доступу: https://tproger.ru/articles/maven-short-intro/ (дата звернення 24.10.2020).

8. Spring Framework. [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Spring_Framework (дата звернення 24.10.2020).

9. Движок шаблонов Thymeleaf. [Електронний ресурс]. – Режим доступу: https://alexkosarev.name/2017/08/08/thymeleaf-template-engine/ (дата звернення 24.10.2020).

10. Введение в HTML. [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/ru/docs/Learn/HTML/Введение_в_HTML (дата звернення 24.10.2020).

11. Тест средств электронной подписи и шифрования документов. [Електронний ресурс]. – Режим доступу: https://www.anti-malware.ru/compare/test_russian_digital_signature_and_encryption_documents#part1 (дата звернення 24.10.2020).

РОЗДІЛ 5
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної магістерської роботи є дослідження програмних засобів створення електронного цифрового підпису. Робота над проектом проходила у офісному приміщенні.

Для організації роботи над проектом створено таблицю 5.1 – розміри приміщення, таблицю 5.2 - характеристики робочого місця, таблицю 5.3 – аналіз небезпечних і шкідливих виробничих факторів, таблицю 5.4 – норми мікроклімату робочої зони об’єкту та проведені необхідні розрахунки.

5.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [4] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

5.1.1 Правові та організаційні основи охорони праці

Законодавство України про охорону праці є системою взаємозв'язаних нормативних актів, що регулюють відносини у галузі реалізації державної політики щодо правових, соціально-економічних, організаційно-технічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці.

Конституція України – основний закон держави, який регламентує найважливіші з погляду держави суспільні відносини.

5.1.2 Організаційно-технічні заходи з безпеки праці

Технічні заходи - технічні засоби, що забезпечують безпечні і нешкідливі умови праці, та пов'язані з впровадженням нового обладнання, пристроїв і приладів безпеки і безпечною експлуатацією засобів виробництва.

Організаційні заходи:

Контроль за технічним станом обладнання, інструментів, будівель і споруд; контроль за дотриманням вимог нормативних документів з охорони праці; нагляд за обладнанням підвищеної небезпеки; організація навчання.

Санітарно-гігієнічні заходи:

контроль за впливом виробничих факторів на здоров'я працівників;забезпечення санітарно-побутових умов згідно з діючими нормами;планування заходів щодо поліпшення санітарно-гігієнічних умов праці.

Соціально-економічні заходи:

надання пільг і компенсацій працівникам, які працюють зі шкідливими і небезпечними умовами праці; створення умов для економічної зацікавленості роботодавця і працівника у поліпшенні умов і підвищенні безпеки праці.

5.2 Аналіз стану умов праці

Робота над створенням автоматизованої системи перевірки якості JavaScript-коду проходитиме в приміщенні компанії. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп’ютером.

5.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 5.1.

Таблиця 5.1 – Розміри приміщення

Найменування	Значення
Довжина, м	6
Ширина, м	4
Висота, м	2,7
Площа, $м^2$	24
Об’єм, $м^3$	64,8

Розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

5.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за . «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2] і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 5.2 - Характеристики робочого місця

Найменування параметра	Наймен. параметра	Нормативне значення
Висота робочої поверхні, мм	690	680 ÷800
Висота простору для ніг, мм	600	не менше 600
Ширина простору для ніг, мм	580	не менше 500
Глибина простору для ніг, мм	650	не менше 650
Висота поверхні сидіння, мм	470	400 ÷500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	450	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷800

Приміщення кабінету знаходиться на третьому поверсі чотирьох поверхової будівлі і має об'єм 64,8 $м^3$, площу – 24 $м^2$. У цьому кабінеті обладнано два місця праці, укомплектовані ПК.Температура в приміщенні протягом року коливається у межах 25-32°С, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 45 дБА. Система вентилювання приміщення — природна неорганізована, а опалення — централізоване.

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою шести люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним

струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет оснащений переносним вуглекислотним вогнегасником ВВК-5 .

5.2.3 Навантаження та напруженість процесу праці

Щодо характеру організування виконання дипломної роботи, то він підпадає під нав’язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи. Роботу над дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви: для операторів персональних комп’ютерів тривалістю 15 хв через дві години роботи.

5.3 Виробнича санітарія

Виробнича санітарія - це система організаційних заходів і технічних засобів, що запобігають або зменшують вплив на працюючих шкідливих виробничих факторів, які в певних умовах можуть привести до травм або професійних захворювань. Основною метою є зменшення або повне усунення впливу несприятливих 1 шкідливих виробничих факторів на організм людини. Оскільки головним у діяльності з охорони праці є профілактика травматизму, заходів щодо поліпшення умов праці й побуту працюючих.

5.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з електронними пристроями» [3], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ПК з ВДТ і ПП. Основними робочими характеристиками ПК є наступні:

– робоча напруга U = +220В ± 5%;
– робочий струм I = 2А;
– потужність споживання P = 350 Вт.

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 5.3).

Таблиця 5.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
фізичні			
- підвищена температура поверхонь обладнання	експлуатація ПК, принтерів, сканерів чи/або серверного обладнання для роботи	2	ДСН 3.3.6.042-99[1]
- підвищена яскравість світла	порушення умов праці (організації місця прац і-налагодження моніторів)	2	ДСанПіН 3.3.2.007-98[2]
- понижена контрастність	-//-	1	ДСанПіН 3.3.2.007-98[2]
психофізіологічні			
- нервово-психічне перевантаженн**я** (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	3	НПАОП 0.00-7.15-18[3] ДСанПіН 3.3.2.007-98[2]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці - сидіння користувача,) та організації робочого часу - безперервна робота)	2	НПАОП 0.00-7.15-18 [3] ДСанПіН 3.3.2.007-98[2]

5.3.2 Пожежна безпека

Електронно-обчислювальна машина (ЕОМ; далі — комп’ютер) — обладнання з необов’язковими додатковими пристроями (пристрої для друку, сканери, модеми, блоки безперервного живлення та інші спеціальні периферійні пристрої). Відеодисплейний термінал (ВДТ; далі — монітор) — частина електронно-обчислювальної

машини, що містить пристрій для візуального відображення інформації; Периферійні пристрої (далі — ПП) — сукупність необов’язкових додаткових пристроїв, які використовуються в процесі діяльності оператора ЕОМ (клавіатура, маніпулятор «миша», дискова система, звукова система, модем, мікрофон, принтер, сканер тощо).

На підприємстві, де експлуатується комп’ютерна техніка, створюється служба охорони праці згідно з «Типовим положенням про службу охорони праці»[9].

Нормативна база

Перелік нормативно-правових актів, які регулюють це питання, досить широкий. Наприклад, ст. 21 Кодексу законів про працю України визначає обов’язки роботодавця щодо забезпечення працівникам комфортних та безпечних умов праці, а ст. 13Закону України «Про охорону праці» закріплює це право з позиції охорони праці.

Особливості охорони праці при роботі з комп’ютером.

Комп’ютерне обладнання повинні підключатися до електромережі лише за допомогою справних штепсельних з’єднань і електророзеток заводського виготовлення.

У штепсельних з’єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їх конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз’єднання при відключенні має бути зворотним.

Не допускається підключати комп’ютерну техніку до звичайної двопровідної електромережі, зокрема з використанням перехідних пристроїв.

5.3.3 Електробезпека

Основні шкідливі та небезпечні фактори, що можуть впливати на організм людини під час роботи з персональним комп'ютером (ПК), такі, як підвищений рівень електромагнітних випромінювань, підвищений рівень іонізуючих випромінювань, підвищена чи знижена іонізація повітря, підвищена яскравість світла, пряма і відбита блискітливість.

5.4 Гігієнічні вимоги до параметрів виробничого середовища

5.4.1 Параметри мікроклімату

Мікроклімат виробничих приміщень - це сукупність параметрів повітря у виробничому приміщенні, які діють на людину у процесі праці, на його робочому місці, у роб зоні. Робоче

місце - територія постійного або тимчасового знаходження людини у процесі праці. Робоча зона - частина простору робочого місця, обмежене по висоті 2 м від рівня підлоги. Параметри мікроклімату це температура повітря Т, 0° С; відносна вологість Y, % та швидкість руху повітря V, м/с.

Значні коливання параметром мікроклімату можуть привести до порушення терморегуляції організму (здатність організму утримувати постійну температуру), що приводить до порушення системи кровообіг, загальної слабкості. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають «Санітарні норми мікроклімату виробничих приміщень» [1] і наведені в табл. 5.4.

Таблиця 5.4 – Норми мікроклімату робочої зони об’єкту

Період року	Категорія робіт	Температура С0	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 2А	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря .

5.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Розрахунок освітлення.

Для будівель виробництв світловий коефіцієнт приймається в межах 1/6 - 1/10:

$$\sqrt{a^2+b^2}\cdot S_b=(1/8\div 1/10)\cdot S_n \qquad (5.1)$$

де S_b – площа віконних прорізів, м2;

s– площа підлоги, м2.

$$s_n = a \times b = 4 \times 6 = 24 \text{ м}^2$$

$$S_{\text{вік}} = \frac{1}{8} \times 24 = 3 \text{ м}^2$$

Приймаємо 2 вікна площею 6 м^2 кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 6 м, ширина 4 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників п виробляється по формулі (5.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M} \tag{5.2}$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м^2; S = 24 м^2;

Z – поправочний коефіцієнт світильника (Z = 1,15 для ламп розжарювання та ДРЛ; Z = 1,1 для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575;

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400 лм (для ЛБ-80).

Підставивши числові значення у формулу (5.2), отримуємо:

$$n = \frac{300 \times 24 \times 1{,}1 \times 1{,}5}{5400 \times 0{,}575 \times 2} \approx 2.$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

5.4.3 Шум та вібрація, електромагнітне випромінювання

Найшкідливішим фактором виробничого і побутового середовищ є шум. Особливо небезпечним є виробничий шум, дія якого є тривалою, тобто постійно супроводжує виробничий процес.

Виробничий шум – сукупність різноманітних за силою і частою звуків. Джерелом шуму можуть бути двигуни, насоси, вентиляційні пристрої, холодильне обладнання, компресори, деякі технологічні процеси – обрубка металу, його клепання, карбування, штамповка, робота на ткацьких верстатах, випробування двигунів, будівельні роботи. Постійним джерелом шуму є транспорт, особливо міський (залізниця, метро, автомобілі) і на сьогодні на вулицях великих міст рівень шуму досягає 80-90 Дб.

Вібрація та її вплив на організм.

Джерела вібрації на виробництві – це пневмо - та електроінструменти ударної або обертальної дій, машини, які установлені на основі без достатньої амортизаційної прокладки, а також транспортні і сільськогосподарські машини.

Вплив вібрації на організм людини залежить від локальної інтенсивності вібраційних хвиль, що викликає зміни стану тканин і органів (стиснення й розтягнення, скручування й згин, утруднення кровопостачання, посилення або послаблення згортальних властивостей крові та ін). Послабити дію вібрації на людину можна засобами віброгасіння, вібропоглинання та віброізоляції.

Електромагнітні хвилі та їх вплив на організм.

Електромагнітні хвилі різного діапазону частот широко використовують в радіолокації, телебаченні, радіозв'язку, фізіотерапії, для термічної обробки металів, приготування їжі, сушки деревини тощо. Їх джерелами є також високовольтні лінії електропередач, електротранспорт. Електромагнітні поля мають певну потужність, енергію і поширюються у вигляді електромагнітних хвиль. Основним параметрами електромагнітних коливань є:

- довжина хвилі;
- частота коливань;
- швидкість розповсюдження.

За частотою антропогенні електромагнітні випромінювання поділяють на:

- низькочастотні (НЧ, 0,003 Гц-30 кГц);

- радіохвилі високочастотного (ВЧ) діапазону (30кГц-300 мГц);
- радіохвилі ультрависокочастотного (УВЧ) діапазону (30мГц-300 мГц);
- надвисокочастотні (НВЧ) хвилі (30мГц-300 гГц).

5.4.4 Вентилювання

Види вентиляції промислових приміщень

Залежно від способу переміщення повітря промислова вентиляція, як і вентиляція приватного будинку, може бути природною і механічною.

Для вентиляції промислових приміщень застосовують припливну, витяжну або припливно-витяжну механічні системи. На великих виробництвах користуються тільки останнім варіантом. У приватних майстернях і невеликих цехах зазвичай встановлюють витяжну механічну вентиляцію. Приплив свіжого повітря у такому випадку відбувається шляхом аерації або інфільтрації, тобто різновидів природної вентиляції. У невеликому виробничому приміщенні витяжної механічної вентиляції зазвичай достатньо.

5.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Заходи безпеки під час експлуатації персонального комп’ютера та периферійних пристроїв передбачають:

- дотримання заходів електробезпеки;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення).

Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів та постійно стежити за справністю ізоляції електромережі та мережевих кабелів.

Вимоги безпеки при надзвичайних ситуаціях:

1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності.Витягнути вилки з розеток. При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК, знайти місце загоряння і виконати всі можливі заходи для його ліквідації.

2) При замиканні, перевантаженні електричного струму на електричному обладнанні, внаслідок ураження грозової блискавки та ймовірної небезпеки ураженням електричним струмом, приймають наступне:

- попередження замикання здійснюється правильним вибором, монтажем експлуатації мереж;

- застосування захисту схем у вигляді швидкодіючих реле, а також вимикачів, плавких запобіжників, автоматичних вимикачів.

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Загальний опір захисного заземлення визначається за формулою:

$$R_{ззп} = \frac{R_з \cdot R_п}{R_п \cdot n \cdot \eta_з + R_з \cdot \eta_п} \tag{5.3}$$

де Rз - опір заземлення, якими когут бать труби, опори, кути і т.п., Ом;

Rш - опір опори, яке з'єднує заземлювачі, Ом;

п - кількість заземлювачів;

ηз - коефіцієнт екранування заземлювача; приймається в межах 0,2 - 0,9; ηз = 0,5

ηш - коефіцієнт екранування сполучної стійки; приймається в межах 0,1 - 0,7; ηш = 0,3;

Опір заземлення визначається за формулою:

$$R_з = \frac{\rho}{2\pi \cdot l} \cdot \left(\ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right) \tag{5.4}$$

де ρ - питомий опір ґрунту, залежить від типу грунту, Ом·м;

для піску - 400 - 700 Ом·м; приймаємо ρ = 500 Ом·м;

l - довжина заземлювача, м; для труб - 2-3 м; l = 2,5 м;

d - діаметр заземлювача, м; для труб - 0,03-0,05 м; d = 0,04 м;

t - відстань від середини забитого в ґрунт заземлювача до рівня землі, м; t = 3 м.

$$\mathrm{R_z} = \frac{500}{2\times3,14\times2,5}\left(\ln\frac{2\times2,5}{0,04} + \frac{1}{2}\ln\frac{4\times3+2,5}{4\times3-2,5}\right) = 31,8 \times (4,8 + 0,2) = 159 \text{ Ом.}$$

Опір смуги, що з'єднує заземлювачі, визначається за формулою:

$$R_{ш} = \frac{\rho}{2\pi \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot t^1} \quad (5.5)$$

де L - довжина смуги, що з'єднує заземлювачі (м) і приблизно дорівнює периметру будівлі: Пбуд. = 32·2 + 25·2 =114 м; L = 114 м;

b - ширина смуги, м; b = 0,03 м;

t1 - глибина заземлення від рівня землі, м; t1 = 0,4 м.

$$R_n = \frac{500}{2 \times 3{,}14 \times 114} \times \ln \frac{2 \times 114^2}{0{,}03 \times 0{,}4} = 0{,}69 \times 14{,}36 = 9{,}9\text{ , Ом}$$

Кількість заземлювачів захисного заземлення визначається за формулою:

$$n = \frac{2 \cdot R_з}{4 \cdot \eta_з} \quad (5.6)$$

де 4 - допустимий загальний опір, Ом;

2 - коефіцієнт сезонності.

Визначаємо загальний опір захисного заземлення:

$$R_{zzn} = \frac{159 \times 9{,}9}{9{,}9 \times 79 \times 0{,}6 + 159 \times 0{,}4} = \frac{1574{,}1}{532{,}8} = 2{,}9\text{ Ом.}$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: Rззп < 4 Ом.

5.6 Охорона навколишнього природного середовища

Діяльність за темою магістерської роботи, а саме: розробка автоматизованої системи перевірки якості JavaScript-коду, в процесі її виконання впливає на навколишнє природнє середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [5], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [6].Законом України «Про відходи» [7], Законом України «Про охорону атмосферного повітря»[10], Законом України «Про

захист населення і територій від надзвичайних ситуацій техногенного та природного характеру»[11], Водний кодекс України [8].

В процесі діяльності за комп,ютером виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- побутові відходи - IV клас небезпеки
- матеріали пакувальні, що вміщують п/ет, п/пр - IV клас небезпеки
- відходи друкуючих пристроїв - IV клас небезпеки

Загалом відходи сфер виробництва і сфери споживання залежно від фізичних, хімічних і біологічних характеристик усієї маси відходу або окремих його інгредієнтів поділяються на чотири класи небезпеки:

І-й клас — речовини (відходи) надзвичайно небезпечні;

ІІ-й клас — речовини (відходи) високо небезпечні;

ІІІ-й клас — речовини (відходи) помірно небезпечні;

IV-й клас — речовини (відходи) мало небезпечні.

Не допускається змішування відходів різних видів і класів небезпеки з будівельними і побутовими відходами, відходами дерев'яної, металевої, синтетичної тари .

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» [7] повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів.

5.7 Висновки до розділу 5

В даному розділі розроблені рекомендації з охорони праці, техніки безпеки при роботі на комп’ютері. Проведений аналіз умов праці, вплив шкідливих та небезпечних чинників на здоров’я людини. Визначено параметри і характеристики приміщення. Приведені рекомендації щодо організації робочого місця, електробезпеки та пожежної безпеки. Наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики та зроблені висновки щодо екології навколишнього середовища.

5.8 Перелік посилань до розділу 5

1. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99

2. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98

3. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з електронними пристроями Зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за № 508/31960. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18

4. Закон України «Про охорону праці» Вводиться в дію Постановою ВР № 2695-XII від 14.10.92, ВВР, 1992, № 49, ст.669. - Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/2694-12

5. Закон України «Про охорону навколишнього природного середовища».Вводиться в дію Постановою ВР № 4005-XII від 24.02.94, ВВР, 1994, № 27, ст.219. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/4004-12

6. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення» Відомості Верховної Ради України (ВВР), 1994, № 27, ст.218){Вводиться в дію Постановою ВР № 4005-XII від 24.02.94, ВВР, 1994, № 27, ст.219

7. Закон України «Про відходи» Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст.242. Режим доступу: www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр

8. Кодекс водний України Вводиться в дію Постановою ВР № 214/95-ВР від 06.06.95, ВВР, 1995, № 24, ст.190 Режим доступу: https://zakon1.rada.gov.ua/laws/show/213/95-%D0%B2%D1%80

9. НПАОП 0.00-4.35-04 Типове положення про службу охорони праці . Із змінами, внесеними згідно з Наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду N 236 (z1191-07) від 02.10.2007 Наказом Міністерства соціальної політики N 148 (z0236-17) від 31.01.2017 Режим доступу: https://zakon.rada.gov.ua/laws/show/z1526-04

10. Закон України «Про охорону атмосферного повітря» Відомості Верховної Ради України (ВВР), 1992, № 50, ст.678 Режим доступу: https://zakon.rada.gov.ua/laws/show/2707-12

11. Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру». Відомості Верховної Ради України (ВВР), 2000, N 40, ст.337 Режим доступу: https://zakon.rada.gov.ua/laws/show/1809-14#o8

ВИСНОВКИ

У магістерській роботі розглянуто та досліджено концепцію розвитку програмних продуктів для створення ЕЦП.

У першому розділі магістерської роботи проведено огляд та порівняльний аналіз програмних засобів створення електронного цифрового підпису.

Розглянута нормативно-правова база створення ЕЦП.

Наголошено, що відповідно до Закону України «Про електронні довірчі послуги», кваліфікований електронний підпис – вдосконалений електронний підпис, який створюється з використанням засобу кваліфікованого електронного підпису і базується на кваліфікованому сертифікаті відкритого ключа.

Схеми формування ЕЦП, що базуються на шифруванні з відкритим ключем, принципово уразливі.

Ефективність використовуваних на практиці схем формування ЕЦП, заснованих на криптографії з відкритим ключем, з точки зору швидкодії, є досить низькою.

Сучасні практичні реалізації схем ЕЦП є вразливими.

З огляду на бурхливий розвиток обчислювальних потужностей сучасних комп'ютерних систем і математичних методів криптоаналізу, практична схема цифрового підпису повинна гарантувати достатній рівень захисту на роки вперед.

При використанні ЕЦП об'єктом захисту поряд з самим об'єктом є і його ЕЦП.

Серед недоліків існуючих схем формування ЕЦП відмічається:

– повільна робота алгоритмів формування і перевірки підпису;

– обмеження на довжину повідомлення, яке підписується.

Відомі програмні рішення пов'язані з обмеженнями на довжину, розбиття повідомлення на фрагменти і підпис кожного фрагмента.

Однак таке рішення часто неприйнятно для використання на практиці, так як результатом буде збільшення обсягу повідомлення і часу виконання процедур створення і перевірки ЕЦП.

В результаті дослідження намічені подальші шлях удосконалення алгоритмів ЕЦП, направлені на зменшення кількості операцій кодування та збільшення криптографічної стійкості ЕЦП.

У другому розділі магістерської роботи проведено огляд та дослідження алгоритмів та моделей криптосистеми з відкритими ключами.

Серед характеристик криптосистеми з відкритими ключами визначено, що найслабшою ланкою при реалізації симетричних криптосистем в системах захищеного

електронного документообігу, електронних банківських платежів і, особливо, електронної торгівлі є питання розподілу ключів.

Для забезпечення обміну конфіденційною інформацією між двома абонентами телекомунікаційної мережі повинен бути згенерований ключ (можливо, одним з абонентів), а потім по деякому захищеному каналу переданий іншим користувачам (іншому абоненту). Як засіб створення захищеного каналу знову може бути використана асиметрична криптосистема.

Найбільшу гостроту питання розподілу і доставки ключів набуває у разі неможливості наперед описати склад інформаційно-телекомунікаційної мережі.

В СВК для зашифрування не використовуються секретні ключі – вони необхідні тільки при розшифрування

Таким чином, одностороння функція з секретом, що використовується в асиметричній системі, є взаємооднозначною, але разом з тим має властивості необерненості.

Розробники криптосистеми RSA скористалися тим фактом, що знаходження великих простих чисел в обчислювальному відношенні здійснюється достатньо просто, але невідомий алгоритм, що виконує за поліноміальний час розкладання на прості множники великих чисел.

Доведено (теорема Рабіна), що розкриття шифру RSA еквівалентне знаходженню такого розкладу. На відміну від RSA асиметрична криптосистема Ель-Гамаля заснована на проблемі дискретного логарифма.

Для побудови СВК можуть бути використані еліптичні криві – математичні об'єкти, визначені над скінченними полями. Слід відзначити, що багато криптографічних систем, розроблених на основі системи RSA, породжують аналоги на еліптичних кривих.

Визначені шляхи подальших досліджень програмних засобів створення ЕЦП.

У третьому розділі магістерської роботи проведено дослідження тестування чисел на простоту і вибір параметрів алгоритму RSA. При побудові асиметричних криптосистем, а також модифікації з параметрів в ході експлуатації виникає необхідність побудови надвеликих псевдовипадкових простих чисел, що мають ті або інші специфічні властивості.

У багатьох випадках, наприклад, у випадку RSA, великі прості числа є ключовими параметрами.

Відповідні обчислювальні процедури включають в себе алгоритми, що реалізують етап перевірки чисел на простоту. В літературі і криптографічній практиці подібні алгоритми носять отримали назву тестів.

Детерміновані тести дозволяють довести, що число, яке тестується, – просте. Практично застосовувані детерміновані тести здатні дати позитивну відповідь не для кожного простого числа, оскільки використовують лише достатні умови простоти.

Детерміновані тести більш корисні, коли необхідно побудувати випадкове велике просте число, а не перевірити простоту, скажімо, деякого єдиного числа.

Детермінований тест використовується, наприклад, в процедурах обчислення несекретних параметрів цифрового підпису типу Ель-Гамаля, встановлених ГОСТ 34.310.

Для реалізації дослідження розглянуті детерміновані тести при побудові асиметричних криптосистем створення ЕЦП, теорема Демітко, тест на основі малої теореми Ферма, основні властивості псевдопростих чисел, властивості чисел Кармайкла, тест Соловея-Штрассена і Ейлерові псевдопрості числа , приклад побудови сильного простого числа, метод Гордона побудови сильних простих чисел.

У четвертому розділі магістерської роботи проведено дослідження сучасних систем ЕЦП шляхом їх програмного моделювання.

За критеріями зручності програмної реалізації і швидкодії це дозволило визначити , що вони в найбільшій мірі задовольняють сформульованим критеріям і можуть бути рекомендовані для практичного застосування.

В результаті дослідження вироблені практичні рекомендації по довжині блоків хеш-функції і довжині ключа алгоритмів ЕЦП.

Таким чином, дослідження на основі тестів, запропонованих NIST, показало, наскільки недосконалі засоби електронного підпису та шифрування документів.

З цього випливає, що поряд з перевірками коректності інтеграції з криптопровайдером, проведеними для рішень, що забезпечують реалізацію технологій електронного підпису та шифрування (що в свою чергу покладено на плечі регуляторів), величезне значення має і тестування правильності реалізації функцій, для яких це рішення створювалося.

Адже помилка в правильності перевірки статусу сертифіката може спричинити фінансовий і репутаційний збиток як для самого розробника рішення, так і для користувачів, які експлуатують це рішення.

Розглянуті питання щодо охорони праці та безпеки у надзвичайних ситуаціях.

Результати роботи та запропоновані рішення можуть бути використані у навчальному процесі кафедри комп’ютерних наук та інженерії при вивченні дисципліни «Захист інформації в комп’ютерних системах».

ДОДАТОК А

Лістинг програми

```
1  import java.io.*;
2  import java.nio.file.Files;
3  import java.nio.file.Path;
4  import java.util.List;
5  import java.util.Map;
6  import java.util.stream.Collectors;
7  @Controller
8  @RequestMapping("/")
9  public class SigningController {
10 private Model model;
11 public SigningController(){
12 this.model = new Model();
13 }
14 @RequestMapping("")
15 public String home(Map<String, Object> model){
16 return "authorization";
17 }
18 @RequestMapping(value = "", params = {"form"})
19 public String homeFormSpecific(@RequestParam String form,
20 Map<String, Object> model){
21 if(form.equals("create")){
22 model.put("showCreate", true);
23 }
24 else{
25 model.put("showCreate", false);
26 }
27 return "authorization";
28 }
29 @RequestMapping(value = "login", method= RequestMethod.POST)
30 public String login(@RequestParam String username,
31 @RequestParam String password,
```

```
32          Map<String, Object> viewModel){
32          User user = model.findUser(username, password);
34          if(user != null){
35          model.setCurrentUser(user);
36          viewModel.put("currentUsername", username); return "activity";
37          }
38          else{
39          viewModel.put("message", "Користувача з такими даними не знайдено! " +
40          "Спробуйте створити нового користувача");
41          return "authorization";
42          }
43          }
44          @RequestMapping(value = "create", method= RequestMethod.POST)
45          public String createUser(@RequestParam String username,
46          @RequestParam String password,
47          Map<String, Object> viewModel){
48          @RequestMapping(value = "delete", method= RequestMethod.POST)
48          public String removeUser(Map<String, Object> viewModel){
50          model.removeUser(model.getCurrentUser());
51          viewModel.put("message", "Користувача "
52          + model.getCurrentUser().getUsername() + " було видалено із системи!");
53          return "authorization";
54          }
55          @RequestMapping(value = "verify", method= RequestMethod.POST)
56          public String verifySignature(@ModelAttribute("object") Object filebject,
57          @RequestParam String username,
58          @RequestParam("file") MultipartFile file, BindingResult bindingResult,
59          Map<String, Object> viewModel){
60          if(!bindingResult.hasErrors()) {
61          byte[] buffer = new byte[]{0};
62          try {
63          buffer = file.getBytes();
64          }
65          catch(IOException ex){
```

```
66              System.err.println(ex.getMessage());
67             viewModel.put("message", "Документ не був підписаний користувачем " +
68           username); }
69              String text = Model.EMPTY_FILE_MARKER;
70              try (InputStream in = new ByteArrayInputStream(buffer)) {
71              text = new BufferedReader(new InputStreamReader(in))
72              .lines().collect(Collectors.joining("\n"));
73              } catch (IOException ex) {
74              System.err.println(ex.getMessage());
75             viewModel.put("message", "Документ не був підписаний користувачем " +
76           username);
77              }
78              try {
79              String[] signatures = ((String)
80             text.subSequence(text.indexOf(Model.DIGITAL_SIGNATURE_STARTING_MARKE
81             R) +
82              Model.DIGITAL_SIGNATURE_STARTING_MARKER.length()
83              , text.length())).split(Model.DIGITAL_SIGNATURE_CONTINUATION_MARKER);
84              String fileText = (String) text.subSequence(0,
85    text.indexOf(Model.DIGITAL_SIGNATURE_STARTING_MARKER));
86              boolean success = false;
87              for(String signature : signatures){
88              if (Signing.verifySignature(model.findUserUnsafely(username), signature, fileText)) {
89              viewModel.put("message", "Документ був підписаний користувачем " + username);
90              success = true;
91              }
92              }
93              if(!success) {
94             viewModel.put("message", "Документ не був підписаний користувачем " +
95             username);
96              }
97              }
98              catch(Exception e){
99              viewModel.put("message", "Документ не був підписаний користувачем " +
```

```
100          username);
101          }
102          }
103          else {
104         viewModel.put("message", "Документ не був підписаний користувачем " +
105         username);
106          }
107          viewModel.put("currentUsername", model.getCurrentUser().getUsername());
108          return "activity";
109          }
110          @RequestMapping(value = "sign", method= RequestMethod.POST)
111          public String putSignature(@ModelAttribute("object") Object fileObject,
112          @RequestParam String path,
113          @RequestParam("file") MultipartFile file, BindingResult bindingResult,
114          Map<String, Object> viewModel){
115          if(!bindingResult.hasErrors()) {
116          byte[] buffer = new byte[]{0};
117          try {
118          buffer = file.getBytes();
119          }
120          catch(IOException ex){
121          System.err.println(ex.getMessage());
122          viewModel.put("message", "Підписати документ не вдалося :(");
123          }
124          String text = Model.EMPTY_FILE_MARKER;
125          try (InputStream in = new ByteArrayInputStream(buffer)) {
126          text = new BufferedReader(new InputStreamReader(in))
127          .lines().collect(Collectors.joining("\n"));
128          } catch (IOException ex) {
129          System.err.println(ex.getMessage());
130          viewModel.put("message", "Підписати документ не вдалося :(");
131          }
132         try (FileWriter fileWriter = new FileWriter(new File(path + file.getOriginalFilename()),
133         false)) {
```

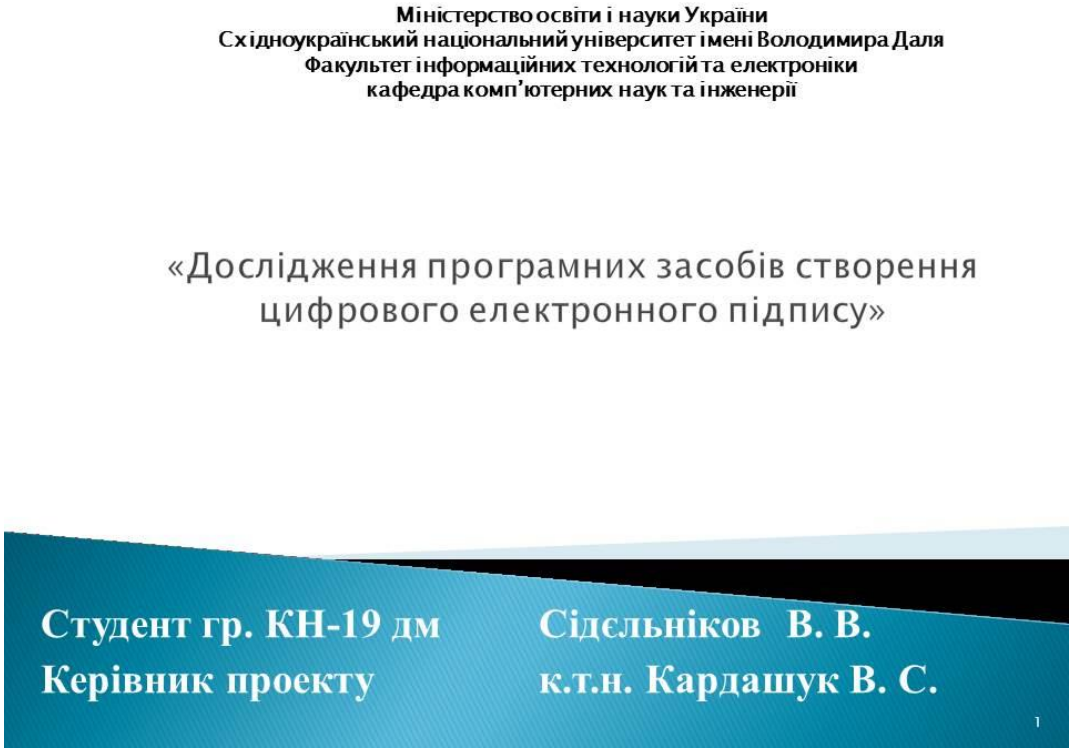
```
134         if(text.contains(Model.DIGITAL_SIGNATURE_STARTING_MARKER)){
135         String originalText = text.substring(0,
136        text.indexOf(Model.DIGITAL_SIGNATURE_STARTING_MARKER));
137                 fileWriter.write(text +
138          Model.DIGITAL_SIGNATURE_CONTINUATION_MARKER
139         + Signing.createSignature(model.getCurrentUser(), originalText));
140         }
141         else {
142         fileWriter.write(text + Model.DIGITAL_SIGNATURE_STARTING_MARKER
143         + Signing.createSignature(model.getCurrentUser(), text));
144         }
145         viewModel.put("message", "Документ було успішно підписано!");
146         } catch (IOException ex) {
147         System.err.println(ex.getMessage());
148         viewModel.put("message", "Підписати документ не вдалося :(");
149         }
150         }
151         else{
152         viewModel.put("message", "Підписати документ не вдалося :(");
153         }
154         viewModel.put("currentUsername", model.getCurrentUser().getUsername());
155         return "activity";
156         }
157         public String getCurrentUsername(){
158         return model.getCurrentUser().getUsername();
159         }
160         } public CustomRSA()
161         {
162         r = new Random();
163         p = BigInteger.probablePrime(bitlength, r);
164         q = BigInteger.probablePrime(bitlength, r);
165         N = p.multiply(q);
166         phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
167         e = BigInteger.probablePrime(bitlength / 2, r);
```

```
168         while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi) < 0)
169         {
170         e.add(BigInteger.ONE);
171         }
172         d = e.modInverse(phi);
173         }
174         public CustomRSA(BigInteger e, BigInteger d, BigInteger N)
175         {
176         this.e = e;
177         this.d = d;
178         this.N = N;
179         }
180         //Тестовий метод для перевірки правильності роботи
181         @SuppressWarnings("deprecation")
182         public static void main(String[] args) throws IOException
183         {
184         CustomRSA rsa = new CustomRSA();
185         DataInputStream in = new DataInputStream(System.in);
186         String teststring;
187         System.out.println("Введіть текст:");
188         teststring = in.readLine();
189         System.out.println("Зашифрований текст: " + teststring);
190         System.out.println("Текст у байтах: "
191         + bytesToString(teststring.getBytes()));
192         byte[] encrypted = rsa.encrypt(teststring.getBytes());
193         byte[] decrypted = rsa.decrypt(encrypted);
194         System.out.println("Розшифрований текст у байтах: " + bytesToString(decrypted));
195         System.out.println("Розшифрований текст: " + new String(decrypted));
196         }
197         private static String bytesToString(byte[] encrypted)
198         {
199         String test = "";
200         for (byte b : encrypted)
201         {
```

```
202         test += Byte.toString(b);
203         }
204         return test;
205         }
206         public byte[] encrypt(byte[] message)
207         {return (new BigInteger(message)).modPow(e, N).toByteArray();
208         }
209         public byte[] decrypt(byte[] message)
210         {
211         return (new BigInteger(message)).modPow(d, N).toByteArray();
212         }
213         }public class Signing {
214         public static String createSignature(User user, String text){
215         try {
216         byte[] data = text.getBytes("UTF8");
217         Signature sig = user.getSignature();
218         sig.initSign(user.getKeyPair().getPrivate());
219         sig.update(data);
220         byte[] signatureBytes = sig.sign();
221         return new BASE64Encoder().encode(signatureBytes);
222         }
223         catch(Exception e){
224         System.out.println(e.getMessage());
225         return null;
226         }
227         }
228         public static boolean verifySignature(User user, String signature, String text){
229         try {
230         byte[] signatureBytes = new BASE64Decoder().decodeBuffer(signature);
231         byte[] data = text.getBytes("UTF8");
232         Signature sig = user.getSignature();
233         sig.initVerify(user.getKeyPair().getPublic());
234         sig.update(data);
235         return sig.verify(signatureBytes);
236         }
237         catch(Exception e){
238         System.out.println(e.getMessage());
239         return false;
240         }
241         }
```

Додаток Б

Презентація



Слайд Б.1

- **Актуальність** – Електронний цифровий підпис (ЕЦП) (англ. Digital signature) за майже сорокарічну історію свого існування пройшов стрімку еволюцію від математичної ідеї У. Діффі і М. Хеллмана, висловленої у 1976 р., до невід'ємного елементу сучасного захищеного мережевого електронного документообігу. ЕЦП – реквізит електронного документа, призначений для захисту електронного документа від підробки або внесення змін, отриманий в результаті криптографічного перетворення інформації з використанням секретного ключа підпису, що дозволяє ідентифікувати власника ключа підпису і встановити відсутність спотворення інформації в електронному документі.

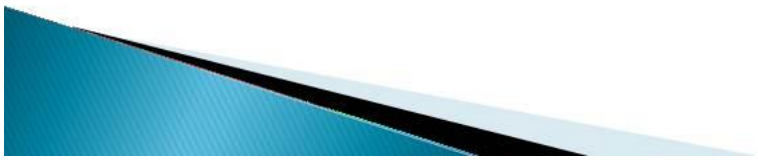
- **Метою роботи** є дослідження найбільш популярних систем ЕЦП, включаючи дослідження застосовуваних хеш-функцій, шляхом їх програмної реалізації з подальшим тестуванням і профілюванням.

- **Об'єкт дослідження** – програмні засоби створення електронного цифрового підпису.

- **Предмет дослідження** – методи криптографічного перетворення інформації з використанням секретного ключа ЕЦП.

- **Методи дослідження** – аналіз існуючих традиційних методів створення та захисту ЕЦП та концептуальних складових інформаційної безпеки.

- **При формалізації задачі дослідження** використано тестування програмних засобів створення ЕЦП та шифрування.

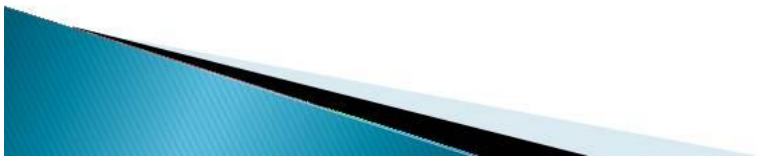


2

Слайд Б.2

Основні задачі магістерської роботи

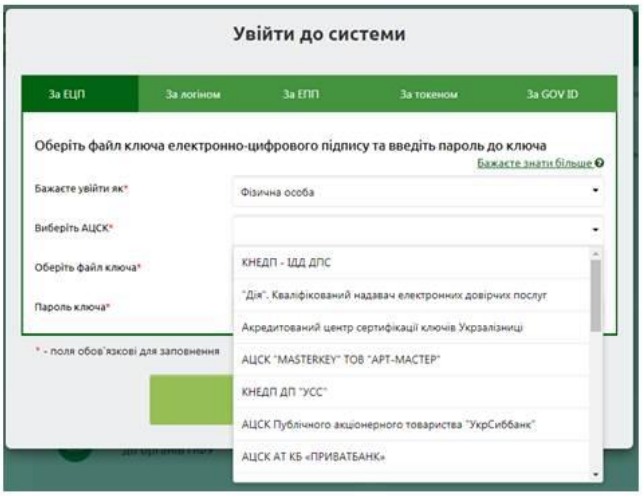
- – дослідження традиційних методів забезпечення безпеки при створенні ЕЦП
- – дослідження існуючих концепцій орієнтованих на безпеку інформації, визначення характеристик та критеріїв підходу для досягнення максимальної ефективності при створенні ЕЦП;
- – розроблення концептуальної основи інформаційно-орієнтованого підходу;
- – дослідження та вибір алгоритмів шифрування та підтримки цілісності даних;
- – дослідження та вибір алгоритмів створення ЕЦП та перевірки аутентифікації;
- – дослідження складової частини інформаційно-орієнтовного підходу, що забезпечує безпечне шифрування даних;
- – тестування програмних засобів створення ЕЦП.



3

Слайд Б.3

АНАЛІЗ ПРЕДСТАВНИЦТВ ЕЛЕКТРОННИХ ПОСЛУГ СТВОРЕННЯ ЕЦП

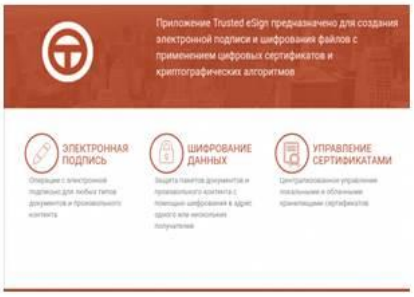


Акредитовані центри сертифікації ключів (АЦСК), утворені відповідно до Закону України «Про електронний цифровий підпис», які надають кваліфіковані електронні довірчі послуги, внесені центральним засвідчувальним органом в Довірчий список як кваліфіковані представники електронних довірчих послуг.

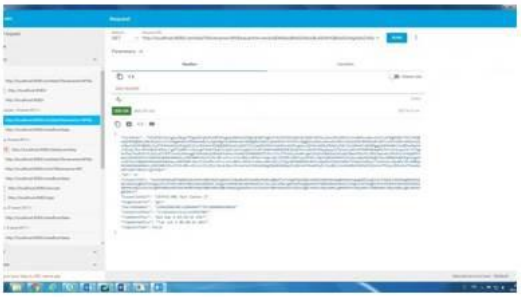
4

Слайд Б.4

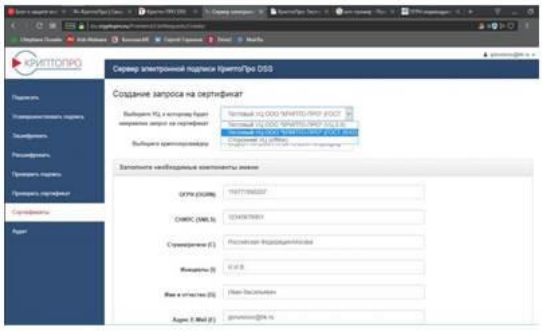
Огляд програмних засобів створення ЕЦП



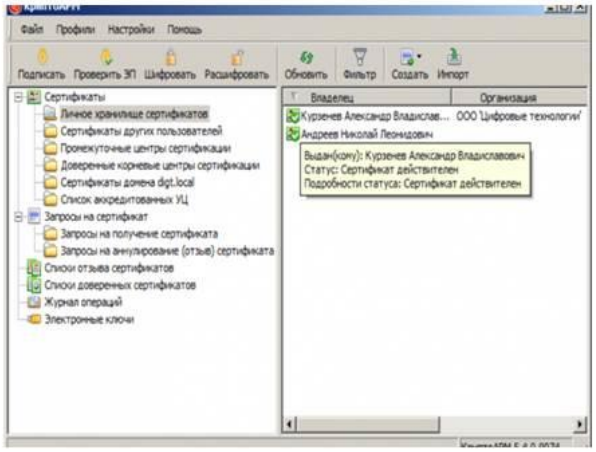
Trusted eSign



Litoria Crypto Platform



КриптоПро DSS



КриптоАРМ

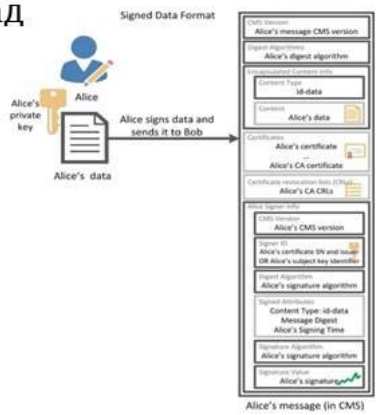
5

Слайд Б.5

Процес створення електронного цифрового підпису

В основі сучасних криптосистем з відкритим ключем обчислювально необернені перетворення частіше за все будуються на основі таких алгоритмічних алгоритмів:

- розкладання великих чисел на прості множники;
- обчислення логарифма в скінченному полі;
- знаходження кратності точки (дискретний логарифм) на еліптичній кривій;
- обчислення коренів алгебраїчних рівнянь над кільцями і полями.

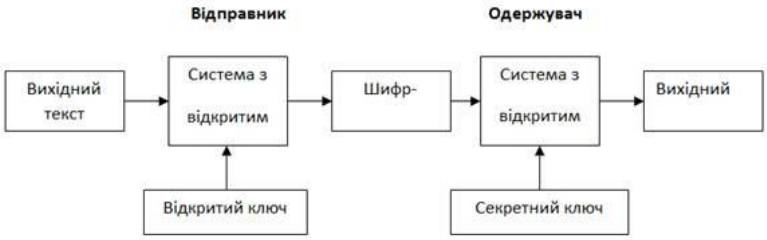


6

Слайд Б.6

АЛГОРИТМИ ТА МОДЕЛІ КРИПТОСИСТЕМИ З ВІДКРИТИМИ КЛЮЧАМИ

Найслабшою ланкою при реалізації симетричних криптосистем в системах захищеного електронного документообігу, електронних банківських платежів і, особливо, електронної торгівлі є питання розподілу ключів.



Для вирішення цієї проблеми на основі результатів, одержаних класичною і сучасною алгеброю, були запропоновані системи з відкритим ключем (СВК).

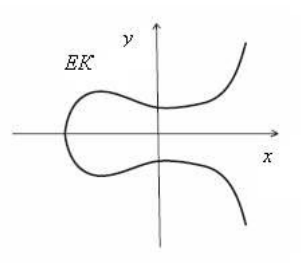
Головним поняттям СВК є однонаправлена функція з секретом (one-way trapdoor function).

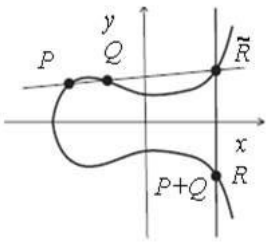
7

Слайд Б.7

Криптосистеми з відкритим ключем, що використовують однонаправлені функції з секретом

1. Діффі і Хеллмана
2. Меркла.
3. Ель-Гамаля.
4. На основі еліптичних кривих.





Слайд Б.8

ДОСЛІДЖЕННЯ ТЕСТУВАННЯ ЧИСЕЛ НА ПРОСТОТУ І ВИБІР ПАРАМЕТРІВ RSA

При побудові асиметричних криптосистем, а також модифікації з параметрів в ході експлуатації виникає необхідність побудови надвеликих псевдовипадкових простих чисел, що мають ті або інші специфічні властивості.

У багатьох випадках, наприклад, у випадку RSA, великі прості числа є ключовими параметрами.

Відповідні обчислювальні процедури включають в себе алгоритми, що реалізують етап перевірки чисел на простоту. В криптографічній практиці подібні алгоритми носять назву тестів.

Детерміновані тести при побудові асиметричних криптосистем створення ЕЦП

Теорема Демітко
Тест на основі малої теореми Ферма
Тест Соловея–Штрассена і Ейлерові псевдопрості числа
Тест Рабіна–Міллера і сильні псевдопрості числа
Метод Гордона побудови сильних простих чисел

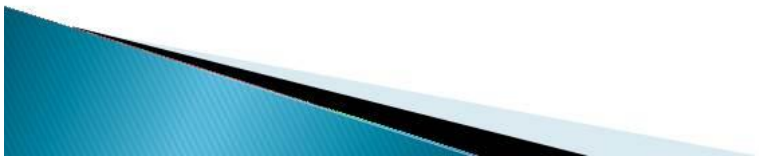
9

Слайд Б.9

Тестування програмних засобів створення ЕЦП та шифрування

Основні характеристики алгоритмів ЕЦП

Алгоритм	Хеш-функція	Рекомендований розмір відкритого ключа, біт	Рекомендований розмір закритого ключа	Рік створення, країна
DSA	SHA-1 або SHA-2	1024-3072	160-256	1994, США
ECDSA	SHA-1 або SHA-2	112-320	80-512	1999, США
ГОСТ Р34.10-2012	ГОСТ Р34.112012	80-320	256-512	2012, РФ
ДСТУ 4145-2002	ГОСТ 34311.95	162-768	256-1024	2002, Україна



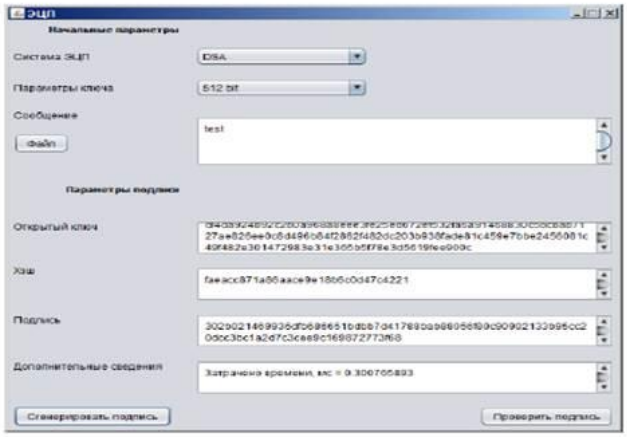
10

Слайд Б.10


Загальна архітектура створеної системи
Управління даними про користувачів
Контролер (Java Spring MVC)
Бізнес-модель
Сутність "Користувач"
Шаблони представлення (Thymeleaf, HTML, CSS)
Реалізація цифрового підпису за допомогою RSA
Видалити свій кабінет
Створити кабінет
Увійти до системи
Увійти до свого кабінету
Вийти зі системи
Actor
Підписати документ
Обрати непідписаний документ
Перевірити підпис документу
Зберегти підписаний документ
Обрати підписаний документ
Діаграма прецедентів системи
11

Слайд Б.11

Основна екранна форма програми



Моделювання ЕЦП в цілому і хеш-функцій проводилося на апаратно-програмних конфігураціях двох типів:

1. Конфігурація А:
CPU: Pentium 987 (ядро Sandy Bridge) 1.5 ГГц (2 МБ L1 cache);
RAM: 4 ГБ DDR3 1300 МГц;
ОС: Win7.

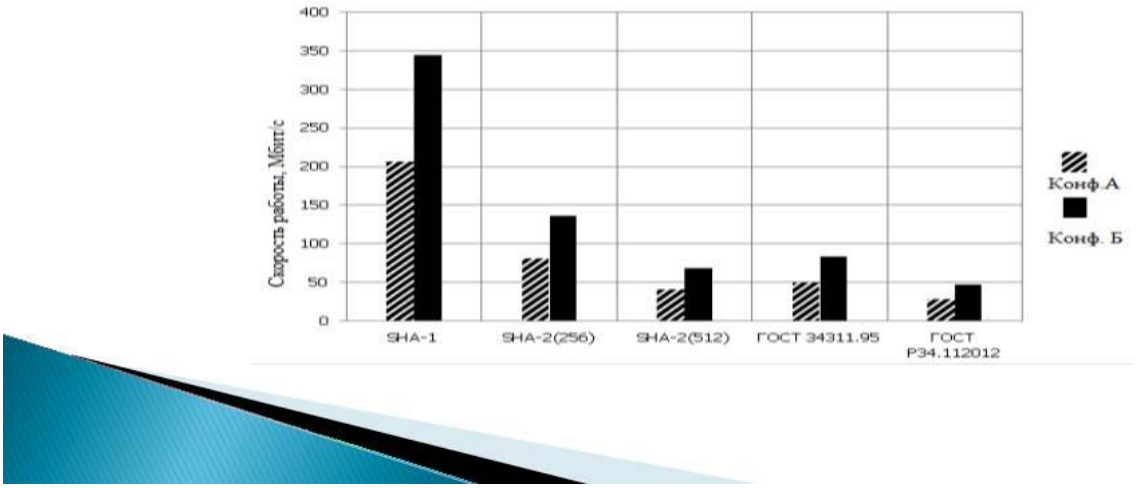
2. Конфігурація Б:
CPU: Core i5 3240T (ядро Ivy Bridge) 2.9 ГГц (3 МБ L1 cache);
RAM: 8 ГБ DDR3 1600 МГц;
ОС: Win7.

12

Слайд Б.12

Результати дослідження швидкодії роботи хеш-функцій

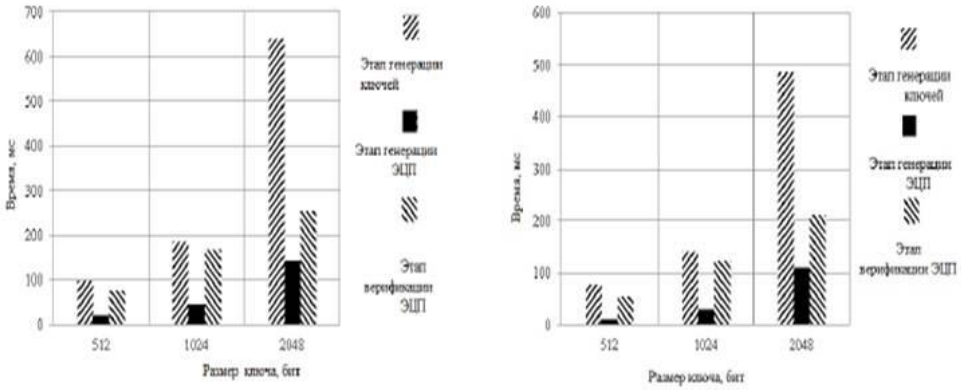
Функція хешування	Кількість раундів	Мова реалізації	Швидкість роботи на конфігурації А, Мбіт/с	Швидкість роботи на конфігурації Б, Мбіт/с
SHA-1	80	Java	206	344
SHA-2 (256)	64	Java	81	135
SHA-2 (512)	64	Java	41	68
ГОСТ 34311.95	256	Java	4928	83
ГОСТ 34.112012	256	Java	2458	46



13

Слайд Б.13

Оцінка швидкодії системи ЕЦП DSA

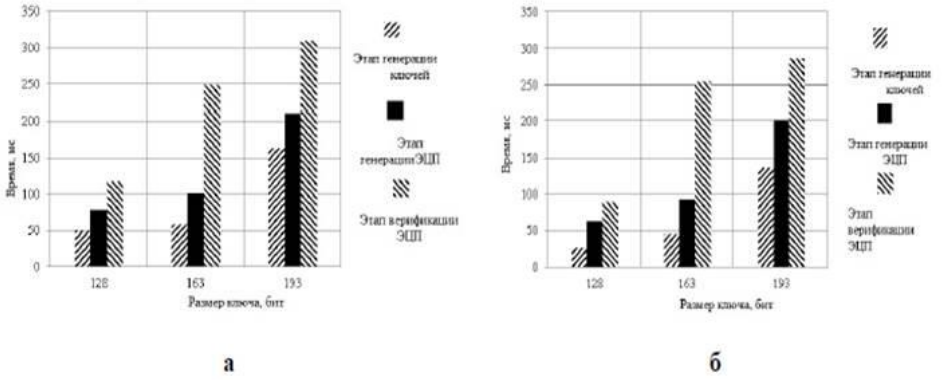


а б

Часовий аналіз етапів DSA в залежності від розміру ключа:
а) конфігурація А, б) конфігурація Б

Слайд Б.14

Оцінка швидкодії систем ЕЦП ECDSA



Часовий аналіз етапів ECDSA в залежності від розміру ключа:
а) конфігурація А, б) конфігурація Б

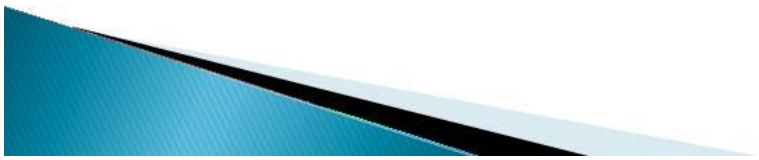
Слайд Б.15

Методологія тестування засобів електронного підпису та шифрування документів

Щоб оцінити роботу представлених програмних засобів, що реалізують технології ЕЦП та шифрування документів, на основі тестів NIST виконано порівняльне тестування коректності їх роботи.

Для порівняння відібрані такі програмні продукти:

1. Litoria Desktop (збірка 1.0.44).
2. КриптоАРМ (збірка 5.4.1.37).
3. Admin-PKI (збірка 5.1.1.1)
4. КАРМА (збірка 56.0.80).
5. КриптоНУЦ (збірка 1.12.2)
6. КріптЕК-Д (демоверсія 1.1.3.42)
7. File- PRO (збірка 2.4.0.15).
8. VipNet CryptoFile (збірка 4.0.1.43722)



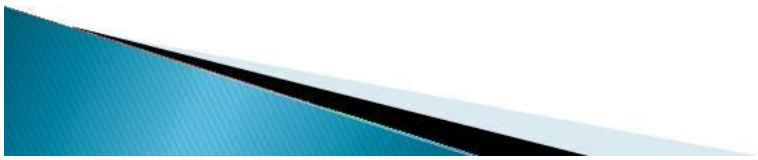
16

Слайд Б.16

Підсумкові результати проходження тестів по NIST

Litoria Desktop	КриптоАРМ	Admin-PKI	КАРМА	КриптоНУЦ
Пройдено тестів				
224	160	173	166	168
Не пройдено тестів (причина – статус сертифікату ЕЦП визначений невірно)				
0	10	18	13	13
Не пройдено тестів (причина – результат невірний)				
0	34	13	25	23
Число нереалізованих тестів				
0	20	20	20	20

Виходячи з результатів тестування безаперечним лідером є програма Litoria Desktop, яка пройшла всі тести по NIST та може бути рекомендована до використання при створенні ЕЦП.



17

Слайд Б.17

Висновки та практичне використання

У магістерській роботі розглянуто та досліджено концепцію розвитку програмних продуктів для створення ЕЦП.

У першому розділі магістерської роботи проведено огляд та порівняльний аналіз програмних засобів створення електронного цифрового підпису. Розглянута нормативно-правова база створення ЕЦП.

У другому розділі магістерської роботи проведено огляд та дослідження алгоритмів та моделей криптосистеми з відкритими ключами.

Серед характеристик криптосистеми з відкритими ключами визначено, що найслабшою ланкою при реалізації симетричних криптосистем в системах захищеного електронного документообігу, електронних банківських платежів і, особливо, електронної торгівлі є питання розподілу ключів. Як засіб створення захищеного каналу знову може бути використана асиметрична криптосистема.

У третьому розділі магістерської роботи проведено дослідження тестування чисел на простоту і вибір параметрів алгоритму RSA.

У четвертому розділі магістерської роботи проведено дослідження сучасних систем ЕЦП шляхом їх програмного моделювання.

За критеріями зручності програмної реалізації і швидкодії це дозволило визначити, що вони в найбільшій мірі задовольняють сформульованим критеріям і можуть бути рекомендовані для практичного застосування.

Результати роботи та запропоновані рішення можуть бути використані у навчальному процесі кафедри комп'ютерних наук та інженерії при вивченні дисципліни «Захист інформації в комп'ютерних системах».

18

Слайд Б.18