Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ПОБУДОВА СИСТЕМИ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ

Текстова частина до магістерської роботи
за спеціальністю „Програмна інженерія” 6.050103

Виконав: студент 2-го року навчання
Баранов Костянтин Олександрович

Керівник Нагірна А.М.,
доцент, канд. фіз.-мат. наук

Магістерська робота захищена
з оцінкою «_____»

Секретар ДЕК_____
«____» _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
доц., канд. фіз.-мат. наук
_____А. М. Нагірна
(підпис)
23 листопада 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на дипломну роботу

студенту 2-го курсу, факультету інформатики
Баранову Костянтину Олександровичу
Розробити Систему електронного цифрового підпису
Зміст ТЧ до магістерської роботи:

Зміст
Анотація
Вступ
1 Теоретичні відомості
2 Огляд проблеми захисту особистого ключа
3 Розробка системи електронного цифрового підпису
Висновки
Список літератури
Додатки

Дата видачі 23 листопада 2020 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Побудова системи електронного цифрового підпису

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту	Термін виконання	Примітки
1	Отримання завдання на дипломну роботу	23.11.2020	
2	Огляд технічної літератури	20.02.2021	
3	Виконати аналіз сучасних систем електронного цифрового підпису	14.03.2021	
4	Проектування системи	17.04.2021	
5	Розробка системи	12.05.2021	
6	Попередній захист роботи	15.05.2021	
7	Аналіз та корегування роботи з науковим керівником	24.05.2021	
8	Написання пояснювальної роботи.	01.06.2021	
9	Створення слайдів для доповіді та написання доповіді	07.06.2021	
10	Захист дипломної роботи	15.06.2021	

ЗМІСТ

АНОТАЦІЯ 5
ВСТУП 6
Розділ 1. Теоретичні відомості 7
1.1 Історичний розвиток криптографії 7
1.2 Криптографічна система 8
1.3 Асиметричні криптографічні системи 9
1.4 Комунікаційна безпека 10
1.5 Хеш-функція 11
1.6 Алгоритм RSA 13
1.7 Електронний цифровий підпис 14
1.8 Сертифікація відкритого ключа 15
Розділ 2. Захист особистого ключа 17
2.1 Обмеження терміну дії сертифікату та його блокування 17
2.2 Шифрування особистого ключа 18
2.3 Smart-картка 19
2.4 Mobile ID 20
2.5 Апаратні модулі безпеки 21
2.6 Висновки за главою 2 22
Розділ 3. Багатофакторний захист особистого ключа 24
3.1 Постановка задача 24
3.2 Опис алгоритму 26
3.3 Технічна реалізація 27
3.4 Демонстрація роботи системи 29
3.5 Висновки за главою 3 33
ВИСНОВКИ 34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 35

АНОТАЦІЯ

Дана робота присвячена створенню системи електронного цифрового підпису, що реалізацією схему захисту особистого паролю з підвищеним рівнем безпеки у масштабних інформаційних системах.

Перший розділ описує основні теоретичні відомості необхідні для побудови систем електронного цифрового підпису. Розглядаються поняття криптографічних перетворень, пояснюється ідея асиметричної криптографії, її головних властивостей.

У другому розділі проводиться порівняльний аналіз найбільш поширених сучасних методів захисту особистого ключа.

У третьому розділі пропонується до розгляду алгоритм захисту особистого ключа з підвищеним рівнем безпеки. Детально описується технічні рішення, які були використанні для побудови системи електронного цифрового підпису, що реалізовує запропонований алгоритм.

ВСТУП

Проблема захисту інформації завжди існувала в інформаційних системах. Сучасні цифрові рішення потребують надійних та ефективних методів забезпечення належного рівня їх безпеки. Системи електронного цифрового підпису, побудовані на підставах криптографічних перетворень, допомагають певній мірі вирішити ці питання, зокрема проблеми цілісності та автентичності даних.

Донедавна криптографічні системи використовувались для забезпечення секретності лише найбільших воєнних та політичних таємниць. Проте все змінилося, і тепер засоби криптографії тісно переплітаються з повсякденним життям пересічної людини. Поява та розвиток систем електронного цифрового підпису надали поштовх до розвитку новітніх економічних, соціальних та адміністративних напрямків інформаційних систем, таких як інтернет-торгівля, онлайн-банкінг чи цифрова держава.

Межі використання систем електронного цифрового підпису значно розширились, а кількість їх користувачів збільшилась на порядок. Тому вимоги до безпеки таких систем теж змінилися. Потрібно зважати на рівень технічної обізнаності користувача, приділити більше увагу на створення простих, прозорих і ефективних механізмів захисту особистого ключа, які органічно втіляться в існуючі технічні рішення та особливості.

Метою даної роботи є побудова системи електронного цифрового підпис з покращеним методом захисту особистого ключа шляхом введення багатофакторної авторизації з урахування особливостей даної предметної області.

1 Теоретичні відомості

1.1 Історичний розвиток криптографії

Криптологія – це наука, що вивчає математичні методи захисту інформації шляхом її перетворення [1]. Криптографія забезпечує безпечний обмін інформацію, доступ до якої можуть отримати лише певні авторизовані суб’єкти.

Умовно становлення криптографії у сучасному виді можна розділити на такі етапи:

а) наївна криптографія;
б) формальна криптографія;
в) наукова криптографія;
г) комп’ютерна криптографія;

У період наївної криптографії текст повідомлень шифрувався зазвичай примітивним способом. В основному використовувалися методи кодування та стетографії для перетворення тексту. Одним із найбільш відомих представником цього періоду є шифр Цезаря, який зводиться до заміни кожного символу вхідного тексту на символ з деким зміщенням відносно поточного по алфавіту.

Період формальної криптографії характеризується появою формалізованих шифрів, стійких до ручного криптоаналізу. Як приклад шифру того часу, можна навести криптографічне перетворення методом багатоалфавітної перестановки. Даний метод був розроблений Леоном Батистом Альбертом, який також є автором першої наукової роботи у галузі криптографії під назвою «Трактат про шифр» датованою 1466 роком. У цей період було сформульовано головний принцип захищеності криптографічних систем, а саме, що безпека шифру має гарантуватися таємністю ключа, а не унікальністю алгоритму.

Етап наукової криптографії розпочався з появою математично доказових криптостійких шифрів.

Сучасний етап, етап комп’ютерної криптографії, характеризується появою криптографічних систем, що реалізуються новітніми цифровими технологіями. Такі шифри здатні здійснювати криптографічні перетворення з криптостійкістю на порядок більшою ніж всі відомі системи до цього. Було засновано новий тип шифрів, які називаються блоковими. Також цей період характеризується відкриттям асиметричних шифрів, що стало справжнім проривом у криптографічній галузі. Сама криптографія, та системи побудованій на досягненнях цією науки, стали активно впроваджуватися в усі сфери діяльності людини: військової, економічної, громадянської та, навіть, особистої.

1.2 Криптографічна система

Криптографічна система – це система, яка забезпечує таємність інформації за допомогою криптографічних перетворень [2]. Основним параметром криптографічних систем є ключ, який дозволяє виконати шифрування та розшифрування даних. В залежності від характеристик ключа, криптографічні системи поділять на симетричні та асиметричні.

Симетричні криптографічні системи використовують один ключ як для операції шифрування, так і для операції розшифрування. Асиметричні ж системи у свою чергу використовують пару різних ключів, які пов’язані між собою математично.

Криптоаналіз – це наука, що вивчає алгоритми дешифрування. Процесом дешифрування називають перетворення зашифрованого повідомлення в вихідне, за умови невідомого ключа розшифрування. Успіх здійснення дешифрування залежить від криптостійкості системи, якою було зашифровано повідомлення. Криптостійкість зазвичай оцінюється в часі, який

необхідний для проведення дешифрування. Значення довжини ключа напряму впливає на криптостійкість.

Сучасні методи забезпечення безпеки криптографічних систем спираються на тезу таємності ключів, при цьому алгоритмічна її складова може бути відома усім учасникам системи без будь-яких негативних наслідків порушення безпеки.

1.3 Асиметричні криптографічні системи

На відмінно від симетричних шифрів, асиметричні використовують два математично пов’язані між собою ключа. Часто один ключ з такої пари називають відкритим, а інший – особистим. Відкритий ключ може бути розповсюджений серед усіх учасників системи, тоді як особистий ключ повинен залишається відомим лише його власнику. Пара ключів для асиметричної криптографічної системи генерується за допомогою спеціального математичного алгоритму, який опирається на проблему вирішення складних обчислювальних задач [4].

Для здійснення передачі зашифрованого повідомлення, відправник має знати відкритий ключ отримувача. Оригінальне повідомлення шифрується відкритим ключем отримувача та надсилається адресату. Отримавши повідомлення, адресат розшифровує його за допомогою особистого ключа. При цьому учасникам системи не потрібно було обмінюватися особистими ключами, як у випадку застосування симетричної криптографії.

Асиметричні системи також надаються можливість здійснення перевірки цілісності повідомлення та перевірки його авторства, що широко використовується для реалізації багатьох прикладних рішень, наприклад, у системах електронного цифрового підпису [5].

Серед недоліків асиметричних криптографічних систем можна зазначити їх низьку продуктивність відносно шифрів симетричного типу, проблему захисту особистого ключа від компрометації, необхідність в створенні централізованих систем для надання достовірної автентифікації, шляхом введення так званих сертифікатів відкритого ключа.

Отже, ключовою відмінністю та перевагою асиметричних криптографічних систем над симетричними полягає у відсутності необхідності в обміні ключами між учасниками системи для здійснення захищеної передачі даних. Особистий ключ використовується для шифрування, а відкритий для розшифрування даних. Також асиметричні криптографічні системи надають можливість для здійснення аутентифікації учасників системи.

1.4 Комунікаційна безпека

Залежно від потреб, канали обміну даними можуть мати різний рівень захисту. Умовно такі комунікаційні системи можна поділити на групи:

а) відкриті;
б) закриті;
в) авторизовані;
г) захищені;

Відкриті комунікаційні системи не гарантують секретність переданих даних та не надають механізмів аутентифікації відправника.

Група закритих комунікаційних систем характеризується забезпеченням секретності переданої інформації. Аутентифікація ж учасників системи залишається нездійсненною. Для реалізації систем цієї групи допустимо використання асиметричних та симетричних криптосистем, або їх комбінації. У такому випадку шифрування повідомлень відбувається з застосуванням

симетричного шифру, а для початкового обміну ключами використовується асиметричне перетворення.

Комунікаційні системи, які входять до авторизованої групи, забезпечують механізм аутентифікації учасників системи, проте секретність повідомлень не передбачається, а інколи навіть забороняється, для дотримання специфічних вимог інформаційної системи. Аутентифікацію можна забезпечити шляхом використання асиметричних криптографічних систем.

Четверта захищена група підтримує одночасно секретність та аутентифікацію. Такі канали зв’язку важливі у сферах діяльності, де можливість встановлення відправника та забезпечення цілісності даних є критичними, наприклад, у комерційних інформаційних системах чи система електронного цифрового підпису.

Системи цієї групи вимагають функціонування централізованої довіреної системи сертифікації. Так вирішується проблема аутентифікації учасників комунікації. Без цієї системи відправник може бути впевнений лише в тому, що повідомлення зашифровані його відкритим ключем призначаються саме йому, проте не можна цілком гарантувати з ким він спілкується, так як публічний ключ відправника залишається анонімним.

1.5 Хеш-функція

Задача хешування полягає у перетворенні довільного розміру масиву вхідних даних у вихідну бітову послідовність фіксованого розміру [3]. Такі перетворення називають хеш-функціями, а результат їх роботи – хеш-кодом. Припускається, що функція хешування є ефективною, так як вона повинна мати змогу обробляти великі обсяги інформація за відносно невеликий час. Щоб хеш-функція була безпечною, вона повинна підтримувати декілька властивостей.

Перша властивість говорить, що функція хешування повинна бути односторонньою, тобто такою, що неможливо було б знайти значення вхідних даних, маючи лише результат роботи такої функції.

Друга властивість дозволяє унеможливлювати знаходження різних повідомлень з еквівалентним хеш-кодом. Хоча функція-хешування може продукувати значення, що викликають колізію, але саме цілеспрямоване знаходження таких повідомлень має займати нераціональну кількість ресурсів.

Хеш-функції широко використовуються в криптографічних системах для забезпечення захисту інформації та безпеки інформаційної системи.

Захист персональної інформації є одним з найбільш поширених випадків використання хеш-функції, наприклад, збереження паролів користувача у секретності. Якщо зберігати паролі учасників системи у відкритому виді, то у разі компрометації цих даних під ударом може опинитися велика кількість користувачів. Рішення проблеми зводиться до зберігання хеш-коду паролю, а не його з значення. При авторизації, пароль, надісланий користувачем, трансформується у хеш-код та зрівнюються з тим, що зберігається на стороні сервісу. Таким чином забезпечується секретність та захист інформації за допомогою хеш-функції.

Також хеш-функції знайшли своє застосування у системах електронного цифрового підпису. Хоча системи асиметричного шифрування дозволяють зашифрувати все повідомлення, авторство та цілісність якого має бути підтвердження, виникає проблема ефективності здійснення операцій шифрування та розшифрування, які потребують більшого часу відносно аналогічних операцій у симетричних криптосистемах.

Вирішення цієї проблеми полягає у застосуванні хеш-функцій. У такому випадку шифрується не весь документ, а лише його хеш-код. Потім оригінальний документ та підписаний хеш-код відправляються адресату. Він розшифровує хеш-код за допомогою відкритого ключа відправника, та

зрівнює отриманий хеш-код з хеш-кодом, який він згенерував самостійно. При умові ідентичності отриманих значень, можна вважати, що дані валідні.

1.6 Алгоритм RSA

Алгоритм RSA – це асиметрична криптографічна система, яка призначена для реалізації безпечної передачі даних [7]. Він включає в себе два фундаментальних положення. По-перше, реалізує криптографічну систему з особистим та відкритим ключами, яка відкидає необхідність обміну секретними ключами для здійснення безпечної передачі даних. По-друге, алгоритм дозволяє провести автентифікацію учасників системи, перевірку можна здійснити маючи відкритий ключ автора повідомлення.

Проста схема довіреного обміну повідомленнями, при застосуванні даного алгоритму, виглядає наступним чином:

1) генерування ключів;
2) шифрування та відправка повідомлення;
3) розшифрування повідомлення;

Генерація ключів виглядає наступним чином:

1) обирається два великих числа p та q;
2) вираховується $n = pq$;
3) вираховується $f = (p - 1)(q - 1)$;
4) обирається таке d, що взаємно просте з f;
5) вираховується $c = d^{-1} \bmod f$;
6) число c зберігається як секретний ключ;
7) числа n та d формують відкритий ключ;

Довжина числа n зазвичай знаходиться в межах діапазону від 200 до 300 символів.

Файл шифрується за допомогою секретного ключа відправника та надсилається адресату, який розшифровує отримане повідомлення відкритим ключем відправника.

Недоліками алгоритму RSA можна назвати низьку швидкодію функцій перетворень та проблема породження підписів дуже великої довжини.

1.7 Електронний цифровий підпис

Електронний цифровий підпис – це набір алгоритмів та технологій для забезпечення автентифікації автора та гарантування цілісності інформації, представленої у цифровому виді [6]. Електронний цифровий підпис реалізовується за допомогою технологій асиметричного шифрування та хеш-функції.

Для створення електронного цифрового підпису необхідно розрахувати хеш-код документу, який необхідно підписати, потім даний хеш-код шифрується приватним ключем автора. Отримане значення вважається електронним цифровим підписом. Щоб перевірити підпис необхідно порівняти хеш-код оригінального документу та значення хешу, отриманим у результаті розшифрування підпису відкритим ключем його автора.

Електронний цифровий підпис дозволяє здійснювати контроль над наступними аспектами електронного обігу інформації:

а) автентичність;
б) цілісність;
в) невідмовність;

Автентичність гарантує автентифікацію власника повідомлення. За необхідності можна змінювати вимоги до набору даних, що можуть служити доказом авторства документу. Набір таких даних визначається користувачами інформаційної системи, необхідних для забезпечення спеціальних процесів.

Цілісність дозволяє перевірити повноту та точність переданої інформації. Забезпечення цілісності свідчить, що інформація не була змінена неавторизованими третіми особами. Інформація має бути захищена від спотворення при здійсненні передачі даних, чи у разі навмисних дій інших учасників системи.

Невідмовність забезпечує неможливість заперечення авторства раніше створеного повідомлення автора. Невідмовність забезпечує безпеку інформаційної системи та захищає права її учасників.

Варто звернути увагу, що дані функції електронного цифрового підпису, мають місце бути за умови, що у системі відсутні скомпрометовані особисті ключі. В іншому випадку безпека інформаційної системи знаходить під загрозою, так як властивості електронного цифрового підпису перестають виконуватися [8].

1.8 Сертифікація відкритого ключа

Управління ключами у великих інформаційних системах запобігає порушенню її безпеки та надає можливість проведення автентифікації учасників системи. Система управління ключами повинна передбачати та забороняти зловмисне використання особистих ключів у разі їх компрометації. Поняття управління ключами включає в себе процеси генерування, зберігання та розподілу ключів.

Часто функції управляння ключами виконується певним довіреним центральний органом, який авторизований надавати послуги управління ключами, що забезпечує підтримку цілісності та автентичності повідомлень. Будь-який користувач може безпечно отримати сертифікат, надавши йому інформації про свій відкритий ключ та певний необхідний набір даних для здійснення проведення автентифікації.

Сертифікат формується шляхом шифрування наданих користувачем значенням відкритого ключа та даних автентифікації, і потім може вільно використовуватися іншими учасниками інформаційної системи. Щоб здійснити перевірку сертифікату, користувач може відправити запит у центральну систему, яка розшифрує отриманий сертифікат своїм закритим ключем та здійснить необхідну валідацію, наприклад, перевірить час дії сертифікату та його відсутність у списках скомпрометованих ключів.

2 Захист особистого ключа

2.1 Обмеження терміну дії сертифікату та його блокування

Виданий користувачу сертифікат відкритого ключа обов’язково має мати термін дії. Зазвичай термін дії встановлюють у межах від одного до двох років. Центр сертифікації ключів зобов’язаний своєчасно скасовувати сертифікати, термін дії яких завершився. Електронний підпис, якій здійснений з використанням особистого ключа зі скасованим чи блокованим сертифікатом, не може вважатися дійсним.

Сертифікат відкритого ключа також може бути скасованим або тимчасово заблокованим на прохання власника сертифікату. Власник сертифікату може надати такий запит, наприклад, у разі несанкціонованого доступу до його приватного ключа. Скомпрометований ключ блокується заради уникнення чи мінімізації шкоди, що може бути завдання третіми особами. Сертифікат додається до списку заблокованих сертифікатів, а учасники інформаційної системи повідомляються про факт блокування.

Чинність сертифікату також може призупинитися у разі тимчасового припинення функціювання кваліфікованого надавача електронних довірчих послуг, який видав цей сертифікат. Поновлення тимчасово заблокованого сертифікату відкритого ключа відбувається за запитом його власника.

Недоліком цього підходу є проблема своєчасного виявлення скомпрометованості особистого ключа. Власник сертифікату може бути тривалий час необізнаний у несанкціонованому доступі до особистого ключа третіми особами, що становить істотну загрозу у безпеці інформаційних систем, перш ніж сертифікат буде відкликано.

2.2 Шифрування особистого ключа

Зберігати особистий ключ у чистому вигляді не рекомендується. Отримавши доступ до місця його зберігання, він може бути легко скопійований та використаний без перешкод третіми особами у власних цілях.

Зменшити ризик використання приватного ключа неавторизованими на це користувачами можна завдяки використанню шифрування. У такому випадку особистий ключ буде зберігатися у зашифровуваному виді, а перед тим, як його використати, особистий ключ необхідно буде попередньо розшифрувати [10].

Зазвичай на вхід до алгоритму шифрування особистого ключа подається сам особистий ключ та довільне кодове слово, яке обирається на розсуд власника ключа. Кодове слово має бути стійке до атак повним перебором, тобто мати належну довжину, містити літери різного регістру, цифри та спеціальні символи. З кодового слова та певної додаткової інформації, формується ключ шифрування. Додатковою інформацією у такій ситуації може слугувати, наприклад, ініціалізаційний вектор алгоритму шифрування. Потім здійснюється необхідне криптографічне перетворення над особистим ключем, і його результат записується до файлу. Вмісту файлу додатково містить заголовок, у якому можна знайти інформацію необхідну для розшифрування, наприклад, назву використаного шифру та ініціалізаційний вектор.

Шифр має бути теж стійким до атак повним перебором. Щоб забезпечити захист від таких атак, часто використовуються криптографічні системи, які працюються відносно повільно. Зазвичай такі шифри використовують хеш-функції, які мають додатковий параметер – фактор роботи, змінюючи який, можна впливати на швидкість здійснення хешування. Чим більше це число, тим повільніший буде алгоритм та відповідно більш

стійким до атак повним перебором. Як приклад можна привести алгоритм BCrypt, який спеціально був розроблений повільним [9].

Щоб скористатися особистим ключем у зашифрованому виді, користувачу необхідно надати кодове слово.

Слабким місцем такого методу забезпечення захисту особистого ключа є статичне кодове слово, яке може бути скомпрометоване чи, у разі низької степені його складності чи складності шифру, підібране повним перебором.

2.3 Smart-картка

Зберігання особистого ключа на окремому пристрої є більш безпечне. Проте це має бути не звичайний пристрій, як, наприклад, USB-флеш-накопичувач, а спеціальний інтелектуальний технічний засіб, який було розроблено специфічних цілей. Смарт-картка – це фізичний електронний пристрій авторизації, що використовується для надання доступу до певних ресурсів [11].

Операції підпису документу відбуваються завдяки апаратно-програмним засобам, які розміщені безпосередньо на смарт-картці. Генерація особистого ключа також може здійснюватися виключно набором технічних рішень самої смарт-картки. При цьому особистий ключ буде існувати лише в одному примірнику та не матиме резервних копій. Це вигідно відрізняє такий спосіб від методу, при якому генерування особистого ключа відбувається поза пристроєм. Так як операція генерування та копіювання особистого ключа на смарт-катку збільшує ризики його компрометації.

Смарт-картки додатково захищають паролем. Перед кожним використанням такої картки, користувач має ввести спеціальний код. Це може бути PIN-код, статичний чи динамічний пароль. Задля запобігання викрадення чи перехвату таких пароля, смарт-картка може бути облаштована автономним

засобом вводу. Взагалі смарт-картки можуть включати в себе досить великий набір додаткових апаратно-програмним рішень, задля вирішення спеціальних задач. Так складна інтелектуальна смарт-картка може бути оснащена кнопками, клавіатурою різної конфігурації, дисплеєм різного розміру, сканером відбитку пальця чи динаміком. Такі картки забезпечують інтеграцію таких функцій, як робота з одноразовими паролями, доступом до інформацією про обліковий запис користувача, проведенням необхідних алгоритмічних обчислень, здійснення безпечних транзакцій та авторизації.

Отже, використання смарт-картки робить зберігання особистого ключа більш надійнішим. Оскільки особистий ключ існує лише в одному примірнику та на окремому пристрої, то користувачеві буде простіше відстежити втрату такої картки та вчасно виконати необхідні дії для захисту особистих даних, наприклад, призупинити дію сертифікату відкритого ключа.

Недоліком смарт-карток є вразливість до фізичних ушкоджень, до різного типу атак націлених на модифікацію транзакцій чи викраденню особистої інформації, відсутність єдиного стандарту для забезпечення взаємодії з різними елементами інфраструктури, та певні інформаційні системи та апаратні пристрої можуть взагалі не передбачати інтерфейсу для роботи з смарт-картками.

2.4 Mobile ID

Технологія Mobile ID є свого роду логічним продовженням смарт-карток, у якому контейнером для апаратно-програмного рішення електронного цифрового підпису є мобільний телефон чи SIM-картка. Особистий ключ створюється безпосередньо на самому пристрої [12].

Використання мобільної ідентифікації за рівнем безпеки може не поступатися звичайному рукописному підпису, якщо всі модулі системи мають необхідного рівня сертифікацію [13].

Щоб підписати документ, система надсилає запит на девайс користувача, який, опираючись на дані запиту, може підтвердити свій намір або відмовити системі у здійсненні підпису. Щоб забезпечити належний рівень захисту даних, модуль Mobile ID додатково вимагає вводу PIN-коду. Після успішного вводу кодової фрази, документ підписується особистим ключем та надсилається у відповідь системі.

Переваги Mobile ID:

- відмова у створенні спеціальних пристроїв на користь мобільним телефонам, який де-факто має переважна більшість користувачів;
- підвищений рівень безпеки порівняно з смарт-картками;
- простота використання;
- можливість реалізації більш специфічних процесів автентифікації та електронного цифрового підпису;

Таким чином Mobile ID має ряд переваг перед іншими носіями особистого ключа, так як є більш зрозуміліший користувачу, потребує менших затрат для реалізації інфраструктури з його використанням, є більш безпечніший, дозволяє використовувати повну функціональність мобільних телефонів для забезпечення кращого досвіду користувача.

Серед недоліків використання Mobile ID можна вказати на вразливість до атак типу «людина посередині», а також проблема компрометації PIN-коду, та використання пристрою третіми особами.

2.5 Апаратні модулі безпеки

Апаратний модуль безпеки (Hardware Security Module, HSM) – це пристрій, який дозволяє здійснювати операції генерування, зберігання та управління криптографічними ключами з підвищеним рівнем захисту даних на програмно-апаратному рівні [14].

Використання HSM дозволяє безпечно управляти особистими ключами через API, так як не потребує додаткового вводу PIN-коду чи інших дій для підтвердження операції від власника ключа. Широко використовується для підвищення безпеки критичних автоматизованих інфраструктурних вузлів, наприклад для проведення операцій з кредитними картками.

Доступні хмарні рішення, які дозволяють інтегрувати існуючі інформаційні системи для роботи з HSM. Основні переваги HSM:

- висока продуктивність;
- розвантаження серверів для здійснення криптографічних перетворень;
- операції генерації, збереження та управління ключами відбуваються безпосередньо самим модулем;
- високий рівень безпеки, як на програмному, так і на апаратному рівні;

Серед недоліків можна зазначити високу вартість такого рішення, особливо для великих систем, та необхідність володінням спеціальними знаннями для роботи з такими модулями. Тому використання HSM для пересічного користувача може виявитися не найкращою опцією.

2.6 Висновки за главою 2

Для підтримки функціювання безпечної системи електронного цифрового підпису необхідно забезпечити захист особистих ключів учасників системи від компрометації. Найбільш поширені методи захисту були описані у попередніх підрозділах. Деякі з них не потребуються втручання користувача у процес здійснення підпису, інші ж полягають у додатковій взаємодії з користувачем, наявністю додаткового обладнання та введення деяких обмежень на елементи інформаційної системи, що можуть функціонувати в рамках надання послуг електронного цифрового підпису.

Одним з найбільш перспективним напрямком є Mobile ID, так як дана технологія не потребує великих затрат для розгортання функціювання масштабних інформаційних систем, може використовувати інтерфейс мобільних пристроїв, є простим та зрозумілим у використанні для користувача та має високий рівень захисту [15]. Відстежити несанкціонований доступ до мобільного пристрою або його пропажу простіше ніж забезпечити належний рівень захисту смарт-картці чи запису особисто ключа у файлові системі.

Проте Mobile ID накладає певні обмеження на спосіб генерації та збереження особистого ключа, так як користувачу необхідно мати активну SIM-картку та мобільний пристрій, щоб користуватися цією технологію. Це також додатково передбачає укладення договору з оператором мобільного зв’язку, який надає сертифіковані послуги Mobile ID.

Далі у цій роботі буде наведено алгоритм здійснення підписання документу, який не потребує спеціальних обмежень на зберігання особистого ключа, та намагається забезпечити той рівень захисту, який пропонує технологія Mobile ID.

3 Багатофакторний захист особистого ключа

3.1 Постановка задача

Однією із умов необхідних для забезпечення цілісності системи електронного цифрового підпису є задача захисту особистих ключів учасників системи від компрометації. На сьогодні існують різні рішення для забезпечення належного в тому чи іншому ступеню рівня безпеки особистого ключа в залежності від вимог предметної області. Деякі рішення засновані на використанні звичайного паролю, деякі вимагають складного обладнання (HSM), інші ж пропонують використання спеціалізованих носіїв для зберігання особистих ключів з підвищеним рівнем захисту (смарт-картка, Mobile ID).

Розвиток мережевих технологій надав можливість реалізації багатьох нових успішних інтернет напрямків, таких як онлайн банкінг, онлайн торгівля, онлайн освіта тощо. Переважна більшість з них стали реальними завдяки використанню сучасних здобутків в області криптографії та розвитку систем електронного цифрового підпису. Розглянемо один з найбільш амбіційних проектів сучасності – створення цифрової держави.

Цифрова держава – це інформаційна система, яка дозволяє надавати широкого спектру послуги у найрізноманітніших сферах діяльності держави її громадянам. Наприклад, громадяни держави можуть здійснювати онлайн реєстрацію ФОП, оплату податків чи зберігати деякі документи онлайн, наприклад, водійське посвідчення, без необхідності у піклуванні про фізичну копію. Для підтвердження тієї чи іншої операції, користувач має скористатися системою електронного цифрового підпису. Підписаний особистим ключем користувача документ зберігається в системі, та служить доказом його цілісності та авторства.

Вимоги до такої інформаційної системи досить строгі. З одної сторони платформа має забезпечувати найвищий рівень безпеки, з іншої – повинна

бути зрозуміла, зручна, швидка, націлена на користувачів з різними навичками володіннями сучасними технологіями, без жорстких обмежень на пристрої, через які можна доступитися до послуг сервісу.

Відповідно дані вимоги мають бути висунуті й до системи електронного цифрового підпису, яка є невід'ємною складовою для забезпечення цілісності платформи цифрової держави. Особливу увагу варто приділити способу захисту особистого ключа, так як безпека системи ЕЦП будується на припущенні відсутності скомпрометованих ключів в системі, а користувач, в свою чергу, може взаємодіяти з такою системою лише при наявності такого ключа, тому ця взаємодія має бути максимально простою, прозорою та безпечною.

Отже, особистий ключ у такій системі повинен:

- мати високий рівень захисту;
- бути простим у використанні;
- не прив'язуватися до конкретного способу зберігання;
- мати економічно сприятливу вартість реалізації для його генерування, зберігання та управління ним;

Серед розглянутих у попередньому розділі різних методів управлінням особистого ключа можна зазначити технологія Mobile ID, яка найбільше відповідає висунутим вимогам. Дійсно, при застосуванні Mobile ID досягається належний рівень захисту ключа від компрометації, відсутня необхідність у введені у експлуатацію додаткових апаратних рішень зі сторони користувача, а використання мобільного телефону, у якості носія особистого ключа, забезпечує зрозумілість та простоту.

Проте Mobile ID має встановлює обмеження на спосіб генерації та зберігання особистого ключа, при цьому користувач обов'язково повинен укласти договір з оператором мобільного зв'язку на дану послугу, та зобов'язатися сплачувати певну суму за надання такої послуги.

Отже, для безпечного функціювання інформаційних систем, які використовують технологію ЕЦП, необхідно вирішити проблему ефективного методу зберігання особистого ключа користувача, який забезпечує належний захист ключа, просто використання, легкість впровадження та доступність.

3.2 Опис алгоритму

Пропоную розглянути алгоритм здійснення електронного цифрового підпису, який увібрав у себе переваги Mobile ID, та водночас не накладає жодних обмежень на спосіб зберігання особистого ключа, тобто це може бути як і мобільний пристрій, так і смарт-картка чи будь-який інший фізичний накопичувач. При цьому користувачу не потрібно вкладати ніяких контрактів – генерування, збереження та управління ключами користувач здійснює найбільш зручним для нього способом на вибір.

Для цього необхідно ввести у процес здійснення ЕЦП елемент двофакторної аутентифікації з певними особливостями, які присутні таким системам ЕЦП. Перед застосуванням особистого ключа, користувач має пройти двофакторну авторизацію. Вона може бути здійснена шляхом відправки СМС на номер телефону, вказаний в сертифікаті цього користувача, чи згенерований прямо на довіреному пристрої користувача. Після чого в базі даних системи ЕЦП створюються відповідний запис про здійснення такої перевірки. Ідентифікатор цього запису відправляється назад користувачу у відповідь на успішну аутентифікацію. Власник ключа додає отриманий ідентифікатор до підпису та надсилає підписаний документ отримувачу. На цьому процес підписання завершується. Далі розглянемо більш детально операції підписання та перевірки підпису у такій системі ЕЦП.

Алгоритм створення електронного цифрового підпису документу виглядає наступним чином:

1) обраховується хеш-функція H_D документу D;

2) формується ідентифікатор запису авторизації I_S;
3) хеш H_D та ідентифікатор I_S шифруються особистим ключем, отримане значення вважається підписом документу;
4) документ та підпис тепер можуть бути відправлені отримувачу;

Алгоритм перевірки підпису:

1) отримувач вираховує хеш-функцію H_{DR} отриманого документу D_R;
2) розшифровує підпис за допомогою відкритого ключа відправника;
3) розшифроване значення H_D порівнюється з H_{DR};
4) ідентифікатор I_S надсилається до центральної системи ЕЦП;
5) якщо хеші збігаються та система підтвердила існування ідентифікатора, то такий підпис вважається достовірним;

Отже, дана схема реалізовує здійснення підписання документу електронним цифровим підписом з використанням багатофакторної авторизації. Метод не обмежує користувача у способі зберігання особистого ключа, натомість він може обрати для найбільш оптимальний варіант.

3.3 Технічна реалізація

Реалізації системи електронного підпису полягає у створенні клієнтської та серверної частини. Клієнтська частина надає користувачу зручний інтерфейс для взаємодії з серверною частиною, дозволяє здійснити підпис документу та згенерувати ключі. Серверна частина відповідає за управління сертифікатами відкритого ключа учасників системи, видачу ідентифікаторів авторизації та здійснення перевірки електронних цифрових підписів.

Клієнтська частина реалізована в якості веб-додатку. Для її розробки був використаний фреймворк Angular, який дозволяє швидко створювати безпечні та сучасні single-page застосунки.

Веб-додаток забезпечує наступні функції:

- генерування ключів;
- здійснення підписання документу;
- здійснення перевірки підпису;

Генерування ключів та здійснення підпису відбувається на стороні клієнта задля забезпечення належного рівня безпеки. Отримання ідентифікатора та перевірка електронного цифрового підпису передбачає формування запиту на сервер.

Для реалізації серверної частино було обрано фреймворк Spring Boot. Даний фреймворк добре себе зарекомендував у створенні безпечних та гнучких застосунків. Це досить потужній фреймворк, проте для реалізації даної системи електронного цифрового підпису достатньо обмежитися модулями Web та MongoDB.

Серверна частина здійснює:

- видачу ідентифікаторів аутентифікації;
- управління сертифікатами відкритого ключа;
- перевірку електронних цифрових підписів на валідність;

Багатофакторна аутентифікація здійснюється шляхом надсилання СМС повідомлення власнику відкритого ключа, для якого зареєстрований відповідний сертифікат. Для реалізації цієї функціональності використано сервіс Vonage, який пропонує API для відправки СМС повідомлень.

Для підтримки функціювання серверної частини використана нереляційна база даних MongoDB. У даному випадку її використання зумовлене простотою інтеграції в проект. MongoDB дозволяє досить швидко розробити необхідний мінімальним прототип для повноцінного функціювання системи. У разі масштабування проекту, дана СУБД може бути заміщена на користь MySQL, яка відповідає міжнародним стандартам та має відповідну сертифікацію.

Отже, реалізація системи електронного цифрового підпису має клієнт-серверну архітектуру. Клієнтська частина спроектована у вигляді веб-додатку, який дозволяє користувачу системи згенерувати ключі, підписати документ та здійснити перевірку підпису. Серверна частина відповідає за управління сертифікатами учасників системи, проведення багатофакторної авторизації шляхом надсилання СМС повідомлення, та здійснення перевірки електронного цифрового підпису.

3.4 Демонстрація роботи системи

Веб-додаток системи електронного цифрового підпису має наступні три сторінки:

- сторінка створення підпису документу;
- сторінка перевірки підпису;
- сторінка генерації ключів;

Щоб згенерувати ключі, на сторінці генерації ключів необхідно заповнити форму (рисунок 3.1). Вказані ім’я, фамілія та номер телефону користувача будуть використані для створення сертифікату відкритого ключа, а значенням поля паролю буде захищено приватний ключ. У результаті система створить два файли, що містять особистий та відкритий ключі.

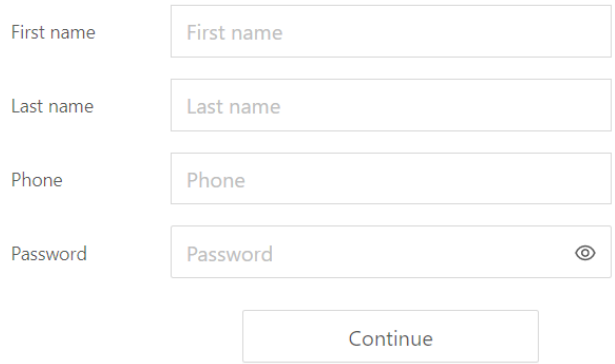

Рисунок 3.1 – Форма для генерації ключів

Використовуючи особистий ключ, можна створити електронний цифровий підпис документу. На сторінці створення підпису документу пропонується виконати дану операцію у кілька кроків. Спершу вказуємо файл особистого ключа та вводимо пароль до нього (рисунок 3.2). Підтверджуємо намір натиском кнопки «Опрацювати ключ».

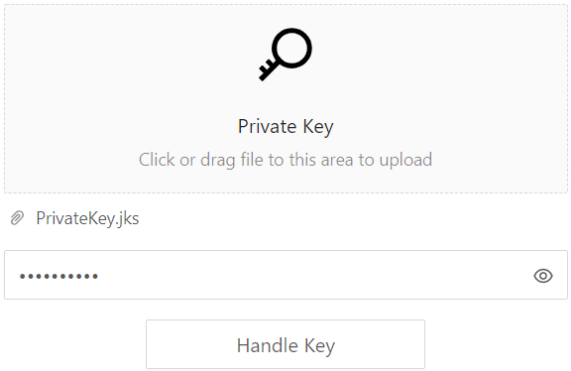

Рисунок 3.2 – Перший крок створення ЕЦП

Потім необхідно вказати документ, для якого ми хочемо створити електронний цифровий підпис (рисунок 3.3). Натискаємо кнопку «Продовжити».

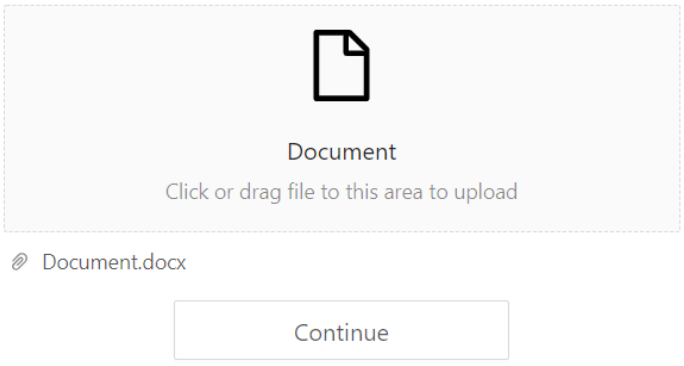

Рисунок 3.3 – Другий крок створення ЕЦП

У межах схеми двофакторної авторизації, система ЕЦП створить тимчасовий код та відправить його на мобільний номер власнику особистого ключа. Отриманий код вводимо у відповідне поле (рисунок 3.4) та натискаємо кнопку «Застосувати підпис».

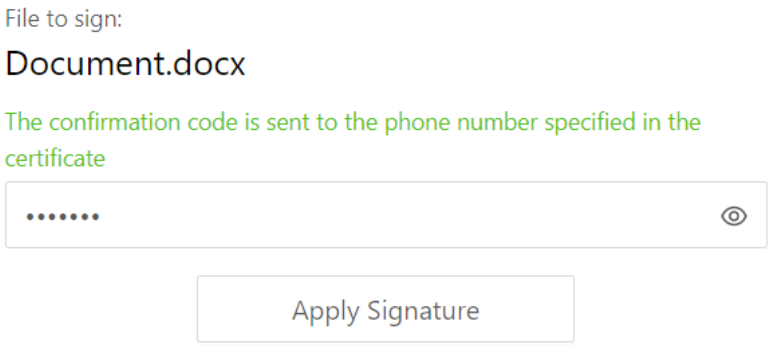

Рисунок 3.4 – Третій крок створення ЕЦП

На четвертому кроці генерується власне підпис документу. Користувач може завантажити файл підпису на особистий пристрій натиснувши кнопку «Завантажити».

Перевірити підпис документу можна на сторінці перевірки підпису. Для цього необхідно вказати оригінальний документ та файл його електронного цифрового підпису (рисунок 3.5).

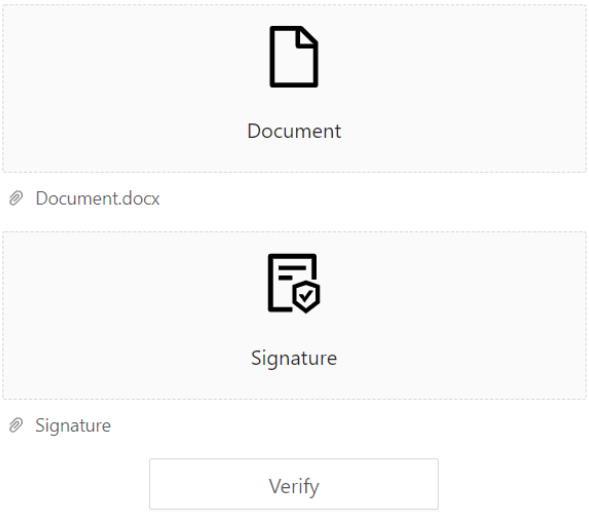

Рисунок 3.5 – Форма для перевірки підпису

Результат перевірки електронного цифрового підпису зображено на рисунку 3.6. Користувачу буде доступна інформація про дату створення підпису та ПІБ його автора.

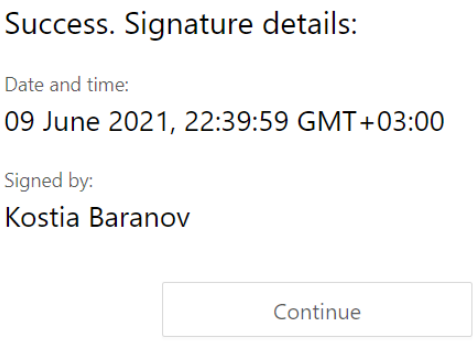

Рисунок 3.6 – Результат здійснення перевірки підпису

3.5 Висновки за главою 3

Запропонований покращений алгоритм створення електронного цифрового підпису документу з підвищеним рівнем захисту особистого ключа полягає у використанні двофакторної авторизації з врахуванням особливостей систем ЕЦП. Здійснення операції підпису передбачає попереднє створення ідентифікатора доступу. Ідентифікатор доступу формується за умови успішного завершення процедури двофакторної авторизації, що полягає в підтвердженні користувачем володінням певного тимчасового секретного коду, який, наприклад, був відправлений йому у форматі СМС повідомлення чи згенерований на довіреному пристрої користувача.

Реалізовано систему ЕЦП у вигляді веб-застосунка з використанням Angular, Spring Boot, MongoDB та Vonage API для відправки СМС повідомлень. Система дозволяє генерувати ключі, створювати та здійснювати перевірку електронного цифрового підпису документу.

ВИСНОВКИ

Було розглянуто основні теоретичні положення криптографії для побудови систем електронного цифрового підпису. Висунуто набір необхідних характеристик для успішної реалізації таких систем.

Проаналізовано сучасні методи забезпечення захисту особистого ключа у системах електронного цифрового підпису. Переглянуті переваги та недоліки їх застосування для вирішення проблеми захисту особистого ключа у масштабних інформаційних системах. Запропоновано покращену схему захисту особистого ключа з використанням багатофакторної авторизації. Виведено набір характеристик систем електронного цифрового підпису, що задовольняють сучасним вимогам безпеки цифрових рішень.

У ході роботи була спроектована та реалізована система електронного цифрового підпису з покращеною схемою захисту особистого ключа, яка забезпечує підвищений рівень безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тарнавський Ю. А. Технології захисту інформації / Юрій Адамович Тарнавський. – Київ, 2018.
2. Антоненко О. Криптографічні методи перетворення інформації / Олександр Антоненко. – Бердянськ: БДПУ, 2015. – 180 с.
3. Steffen T. Cryptographic Hash Functions [Електронний ресурс] / Thomsen Steffen. – 2009. – Режим доступу до ресурсу: https://backend.orbit.dtu.dk/ws/portalfiles/portal/5025771/sst_thesis_v1.0.pdf.
4. Simmons G. Symmetric and Asymmetric Encryption [Електронний ресурс] / Gustavus Simmons // Computing Surveys. – 1979. – Режим доступу до ресурсу: https://www.princeton.edu/~rblee/ELE572Papers/CSurveys_SymmAsymEncrypt-simmons.pdf.
5. Brush K. Asymmetric Cryptography (Public Key Cryptography) [Електронний ресурс] / Kate Brush – Режим доступу до ресурсу: https://searchsecurity.techtarget.com/definition/asymmetric-cryptography.
6. Holguín F. Analysis of digital signature based on the public key infrastructure [Електронний ресурс] / Fresia Holguín. – 2018. – Режим доступу до ресурсу: http://dx.doi.org/10.21503/hamu.v5i2.1622.
7. Milanov E. The RSA Algorithm [Електронний ресурс] / Evgeny Milanov. – 2009. – Режим доступу до ресурсу: https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf.
8. Minimum Requirements for the Issuance and Management of Publicly-Trusted Code Signing Certificates [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://pkic.org/uploads/2016/09/Minimum-requirements-for-the-Issuance-and-Management-of-code-signing.pdf.
9. Provos N. A Future-Adaptable Password Scheme [Електронний ресурс] / N. Provos, D. Mazières // The USENIX Association. – 1999. – Режим доступу до ресурсу: https://www.usenix.org/legacy/events/usenix99/provos/provos.pdf.

10. Kleppmann M. Improving the security of your SSH private key files [Електронний ресурс] / Martin Kleppmann. – 2013. – Режим доступу до ресурсу: https://martin.kleppmann.com/2013/05/24/improving-security-of-ssh-private-keys.html.
11. Smart card [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Smart_card.
12. Mobile signature [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Mobile_signature.
13. Mobile identification: Implementation, challenges, and opportunities [Електронний ресурс] // International Telecommunication Union. – 2017. – Режим доступу до ресурсу: https://www.itu.int/en/ITU-D/Regional-Presence/Europe/Documents/Publications/2f_MID_Report_415270-BAT3.pdf.
14. Hardware security module [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Hardware_security_module.
15. Mobile ID [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://idev-hub.com/uk/mobile-id-bezpechnyi-dostup-ukraine-in-smartphone/.