https://doi.org/10.31891/2307-5732-2025-359-127
УДК 621.391:004.056.5

РУДИЙ РОМАН
Чернівецький національний університет імені Юрія Федьковича
https://orcid.org/0009-0002-5161-5032
e-mail: rudyi.roman@chnu.edu.ua
ГРЕСЬ ОЛЕКСАНДР
Чернівецький національний університет імені Юрія Федьковича
https://orcid.org/0000-0002-8465-193X
e-mail:.o.hres@chnu.edu.ua

МЕТОДИКА ВАРІАТИВНОГО ІНДЕКСУВАННЯ РІВНЯ КОНФІДЕНЦІЙНОСТІ ТА ЗАХИСТУ ПЕРСОНІФІКОВАНИХ ДАНИХ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ І МЕРЕЖАХ

У статті проведено комплексний аналіз сучасних способів зберігання персоніфікованих цифрових даних та виконано їх класифікацію за типами пристроїв і архітектурними особливостями. Розглянуто основні підходи до організації систем зберігання інформації з урахуванням вимог до безпеки, надійності та ефективності використання апаратних ресурсів. Запропоновано архітектуру та узагальнену модель пристрою зберігання даних, орієнтовану на забезпечення багаторівневого доступу до конфіденційної інформації. Обґрунтовано вибір формату представлення даних і методів їх захисту. Проведене моделювання підтвердило можливість вирішення низки практично важливих завдань, зокрема аналізу вимог до захисту інформації за основними критеріями інформаційної безпеки: конфіденційністю, цілісністю та доступністю. Запропонований підхід дозволяє здійснювати класифікацію персональних даних залежно від рівня їх важливості та вимог до своєчасності доступу, а також визначати відповідні рівні доступу й терміни зберігання. Це, у свою чергу, забезпечує оптимізацію використання апаратних ресурсів системи та підвищує ефективність процесів накопичення персональної інформації. Розроблена прикладна програма має демонстраційний характер і слугує підтвердженням працездатності запропонованої моделі. Подальші дослідження доцільно спрямувати на розроблення методів автоматизованого сортування персональних даних за визначеними критеріями, а також на реалізацію проаналізованих підходів із використанням програмно та апаратно реконфігурованих ресурсів, що дозволить підвищити гнучкість і масштабованість систем зберігання даних.

Ключові слова: персоніфіковані дані, зберігання інформації, архітектура сховищ, інформаційна безпека, багаторівневий доступ, класифікація даних, оптимізація ресурсів.

RUDYI ROMAN, HRES OLEKSANDR
Yuriy Fedkovych Chernivtsi National University

METHOD OF VARIABLE INDEXING OF THE LEVEL OF CONFIDENTIALITY AND PROTECTION OF PERSONALIZED DATA IN TELECOMMUNICATION SYSTEMS AND NETWORKS

The article provides a comprehensive analysis of modern methods of storing personalized digital data and their classification by device types and architectural features. The main approaches to organizing information storage systems are considered, taking into account the requirements for security, reliability, and efficiency of hardware resource use. The architecture and generalized model of a data storage device are proposed, focused on ensuring multi-level access to confidential information. The choice of data representation format and methods for their protection is justified. The modeling confirmed the possibility of solving a number of practically important tasks, in particular, the analysis of information protection requirements according to the main criteria of information security: confidentiality, integrity, and availability. The proposed approach allows for the classification of personal data depending on their level of importance and requirements for timeliness of access, as well as determining the appropriate access levels and storage periods. This, in turn, ensures optimization of the use of system hardware resources and increases the efficiency of personal information accumulation processes. The developed application is of a demonstration nature and serves as confirmation of the operability of the proposed model. Further research should be directed towards developing methods for automated sorting of personal data according to specific criteria, as well as implementing the analyzed approaches using software and hardware reconfigurable resources, which will increase the flexibility and scalability of data storage systems.

Keywords: personalized data, information storage, storage architecture, information security, multi-level access, data classification, resource optimization.

Стаття надійшла до редакції / Received 03.10.2025
Прийнята до друку / Accepted 15.11.2025

Постановка проблеми

Здебільшого сучасні інфокомунікаційні системи можна описати термінами глобалізації та індивідуалізації. З одного боку, інфокомунікаційні системи дозволяють нам мати доступ до будь-якої інформації у будь-якій точці Землі. І це можна характеризувати як глобалізацію доступу до інформації. З іншого боку, кожна людина може отримувати інформацію практично звідусіль, і відповідно цей процес можна описати як індивідуалізацію доступу до інформації. Проте збільшення можливостей інфокомунікаційних систем щодо обміну інформації також призводить до негативних наслідків. В першу чергу – це необхідність захисту інформації та інформаційних систем, захисту інфокомунікаційних систем як засобів комунікації. Це ж

стосується захисту персональних даних кожного учасника інформаційного поля [1][2].

Поняття персональні дані та персоніфіковані дані в сучасній літературі, наукових дослідженнях, юридичній практиці часто ототожнюють [3][4][5]. Проте, на нашу думку, між цими поняттями існує суттєва різниця. Коли ми говоримо про персоніфіковані дані, то маємо на увазі, що це є інформація, відомості, які стосуються конкретної людини, яка може бути ідентифікована за певними фізичними чи фізіологічними показниками, отриманими з цих даних. Якщо ж мова йде про персональні дані, то ця інформація вже відноситься до конкретної діяльності людини, і ця інформація може підлягати обміну, використанню для спілкування, застосуванню в роботі і тому подібне. Як в одному, так і в іншому випадку вимоги до захищеності, доступу, цілісності інформації можуть бути різними. Якщо ми говоримо про персоніфіковану інформацію, то це, наприклад, можуть бути медичні дані особистості, доступ до яких має обмежене коло користувачів: сімейний лікар і пацієнт. При необхідності надання спеціалізованої допомоги необхідно надати доступ третій особі – профільному лікарю.

Аналогічно, якщо особа володіє певною корпоративною інформацією, працює над певним проектом, то ця інформація може бути закритою для сторонніх осіб. Однак, коли проект виходить на завершення і представляється замовнику, тоді коло осіб може розширюватися.

Таким чином, як в першому, так і другому випадку, рівень конфіденційності інформації може змінюватися. Разом з цим можуть змінюватися і такі показники, як цілісність інформації і доступність інформації, оскільки в певних конкретних випадках ця інформація може дозовано надаватися третім особам. Загалом, тріада КДЦ (конфіденційність, доступність, цілісність), як і гексада Паркера (КДЦ + автентичність, корисність, володіння) можуть бути показниками порушень доступу не тільки до інформаційних систем в цілому, а й до персональних чи персоніфікованих даних зокрема.

Варто відмітити, що зміна статусу конфіденційності можлива і користувачем інформації на різних етапах її використання і може оцінюватись наприклад показниками «важливості» і «своєчасності». Тому актуальним є питання контролю і управління рівнем КДЦ персоніфікованої інформації у процесі її використання протягом життєвого циклу – від генерування даних до їх видалення.

Метою даного дослідження було обґрунтування уніфікованого підходу до визначення і класифікації статусу конфіденційності інформації за критеріями «важливості» і «своєчасності», та можливості змінювати рівні захисту персоніфікованих даних у залежності від вимог до тріади КДЦ. При цьому «рівні захисту» ми розуміємо як апаратно-програмні можливості ускладнення зловмисного порушення цілісності чи розкривання сутності конфіденційної інформації.

Аналіз досліджень та публікацій

Критерії «важливості» і «своєчасності» можуть оцінюватись за різною шкалою. Зокрема це можуть бути бальні шкали різної градації з присвоєнням значень від «непотрібно – неважливо – потрібно – важливо – дуже важливо» із цифровою відповідністю від 1 до 5. Критерій «своєчасності» може приймати значення «втрачена актуальність (підлягає видаленню) – актуально (зберігати) – довготривала актуальність (архівувати)».

Зберігання цифрових даних полягає у фіксації інформації в цифровому форматі на спеціальних носіях, зазвичай із використанням електронних технологій та спеціального апаратного та/чи програмного забезпечення для обмеження прав доступу сторонніх осіб за рахунок шифрування або захисного кодування. Пристрої пам’яті надають можливість користувачам накопичувати значні обсяги інформації, займаючи при цьому мінімальний фізичний простір, а також спрощують передавання даних між користувачами. Збереження інформації може мати як тимчасовий, так і довготривалий характер. Цифрові засоби зберігання також можуть забезпечувати різну «глибину доступу», а відповідно і різні рівні захисту персоніфікованих даних як на програмному рівні (за рахунок вкладених папок), так і на апаратному (за рахунок опосередкованого кілька крокового доступу до ресурсів. іОкрім цього, носії інформації часто використовуються для створення резервних копій важливих файлів. Водночас зберігання інформації є необхідним для захисту даних від втрати внаслідок стихійних лих, збоїв програмного забезпечення чи шахрайських дій. Саме тому цифрові сховища також виконують функцію надійного інструменту резервного копіювання.

Пристрої для зберігання інформації зазвичай поділяють на дві ключові групи: системи безпосереднього доступу до даних та мережеві сховища. Безпосереднє зберігання, яке також називають сховищем із прямим підключенням (Direct Attached Storage, DAS), розташовується поруч із комп’ютером і під’єднується до нього безпосередньо. У більшості випадків це окремий фізичний пристрій, що забезпечує доступ до даних лише для одного комп’ютера. Окрім зберігання інформації, DAS може використовуватися для локального резервного копіювання, однак можливості спільного доступу до таких систем є обмеженими [6]. До категорії пристроїв DAS належать дискети, оптичні носії, жорсткі диски, флеш-накопичувачі та твердотільні накопичувачі (SSD).

Мережеві сховища інформації забезпечують доступ до даних одразу для кількох комп’ютерних систем шляхом використання мережевої інфраструктури, що робить їх доцільними для спільного використання ресурсів і колективної роботи з файлами. Такі рішення є більш надійними для організації резервного копіювання та підвищення рівня захисту інформації. Найбільш розповсюдженими архітектурами мережевого зберігання є мережева система зберігання даних (NAS) та мережа зберігання даних (SAN). Система NAS зазвичай реалізується у вигляді окремого автономного пристрою, який об’єднує кілька накопичувачів у відмовостійкий масив для зберігання даних. На відміну від неї, SAN являє собою складну інфраструктуру, що

може включати різноманітні типи пристроїв зберігання, зокрема твердотільні накопичувачі, флеш-носії, гібридні системи та інтегровані хмарні сховища [7, 8].

Цифрові носії для зберігання інформації зазвичай класифікують на такі основні типи:

- твердотільні накопичувачі та флеш-носії;
- гібридні системи зберігання;
- хмарні сервіси зберігання даних;
- гібридні хмарні рішення.

Флеш-пам'ять належить до категорії енергонезалежної довготривалої пам'яті та використовується для запису й збереження цифрових даних. Твердотільні накопичувачі (SSD) базуються на технології флеш-пам'яті, що дозволяє зберігати інформацію без використання механічних компонентів. На відміну від традиційних жорстких дисків, SSD не містять рухомих елементів, що забезпечує менший час доступу до даних, а також подовжити шлях доступу при застосування на проміжному етапі мікроконтролер чи інший пристрій для комунікації сховища даних із робочим місцем користувача.

Використання твердотільних накопичувачів і флеш-пам'яті дає змогу досягти вищої швидкості передавання даних порівняно з класичними жорсткими дисками. Водночас системи зберігання, побудовані виключно на флеш-технологіях, можуть мати значну вартість. Саме тому на практиці часто застосовується гібридний підхід, який поєднує високу продуктивність флеш-накопичувачів із великою місткістю жорстких дисків. Такий збалансований підхід до побудови інфраструктури дозволяє ефективно використовувати різні технології відповідно до конкретних завдань зберігання інформації.

Хмарні сховища виступають доступною та масштабованою альтернативою локальному зберіганню даних на фізичних носіях або в мережевих системах. Провайдери хмарних сервісів забезпечують розміщення інформації на віддалених серверах із можливістю доступу до неї через мережу Інтернет або захищене приватне з'єднання. При цьому постачальник послуг бере на себе відповідальність за розміщення, захист, адміністрування та технічне обслуговування серверної інфраструктури, гарантуючи доступність даних у необхідний момент.

Гібридне хмарне сховище є поєднанням можливостей приватних і публічних хмарних середовищ. Використання такого підходу надає гнучкість у виборі місця розміщення інформації залежно від її характеристик та вимог до безпеки [9]. Зокрема, дані з підвищеним рівнем конфіденційності, до яких висуваються жорсткі вимоги щодо захисту, доцільно зберігати в межах приватної хмарної інфраструктури. Водночас менш чутлива інформація може розміщуватися у відкритій хмарі загального користування.

Виклад основного матеріалу. Апаратні рішення

Одним з головних завдань даного дослідження є також моделювання пристрою для демонстрації багаторівневої системи збереження персональної інформації користувачів: персональний комп'ютер – контролер з вбудованою пам'яттю – сховище, наприклад SSD карта. В якості об'єкта з обмеженим доступом є персоніфіковані дані у вигляді облікових записів. Таким чином реалізується трирівнева модель доступу в залежності від «важливості» чи конфіденційності даних. Структурна модель пристрою передбачає наявність таких базових елементів:

- апаратної платформи Arduino;
- засобу для накопичення даних;
- прикладного програмного забезпечення;
- модуля для передавання даних.

Ключовим компонентом апаратної частини є плата Arduino Uno. Платформа Arduino широко використовується в системах збору інформації та керування зовнішніми пристроями. У межах цього проєкту вона виконує функції обробки команд користувача та організації запису інформації на носій зберігання.

Архітектура Arduino передбачає два можливі способи збереження інформації:

- використання вбудованої пам'яті EEPROM;
- застосування зовнішніх пристроїв зберігання даних.

EEPROM належить до енергонезалежних типів пам'яті, що забезпечують збереження даних після вимкнення живлення. Проте обсяг цієї пам'яті в Arduino Uno становить лише 1 КБ, чого недостатньо для зберігання значного обсягу інформації, оскільки така місткість дозволяє розмістити приблизно 128 символів, що відповідає лише кільком записам [10]. З огляду на обмеженість вбудованої пам'яті, більш доцільним є використання зовнішніх носіїв інформації, зокрема карт пам'яті стандартів SD або microSD.

Передавання даних між пристроєм та програмним забезпеченням може здійснюватися двома способами: через дротове з'єднання з використанням USB-інтерфейсу або за допомогою бездротового каналу зв'язку на базі модуля Wi-Fi, якщо використати Arduino сумісний модуль ESP32. Керування пристроєм реалізується через прикладний програмний продукт, що працює під управлінням операційної системи Windows. За допомогою відповідного інтерфейсу додаток передає команди та інформацію на мікроконтролер Arduino, який, у свою чергу, виконує збереження даних на зовнішньому носії. Інформація записується у текстовий файл та структурується у вигляді таблиці з такими полями:

- найменування облікового запису;
- ім'я користувача (логін);
- пароль.

Модулі карт SD і microSD дозволяють спілкуватися з картою пам'яті та записувати або читати інформацію на них. Модулі взаємодіють по протоколу SPI. Щоб використовувати ці модулі з Arduino, потрібна бібліотека SD [13]. Ця бібліотека встановлена в програмі Arduino за замовчуванням. На картах SD і microSD бібліотека підтримує файлові системи FAT16 і FAT32. Ці модулі не можуть працювати з картами пам'яті великої ємності. Зазвичай максимальна ємність карт, що може зчитати модуль, становить 2 ГБ для карт SD і 16 ГБ для карт microSD.

Всі записи зберігаються в одному текстовому файлі, що знаходиться на SD card. При обробці записи зчитуються послідовно. Для цього кожний запис повинен записуватися в новий рядок. Це надасть можливість швидше обробити записи та здійснити пошук потрібного. Також для розділення полів запису використовується пробіл. Персональні дані, що будуть зберігатися на пристрої повинні бути структуровані. Від цього буде залежати швидкодія пристрою. Пристрій зберігатиме інформації про облікові записи користувача, де основними полями є назва облікового запису, логін та пароль. Також можна зберігати записи, де потрібно вказати назву та пароль. Це дозволить розширити функціонал пристрою.

Захист даних є важливою частиною. У наш час існує багато кіберзагроз, ціллю яких стає збір даних. Тому існують різні підходи, щоб запобігти кібератакам чи витоку даних. Для захисту даних на пристрої впроваджено два підходи: авторизація та шифрування деяких полів облікового запису.

Авторизація. Для того, щоб користувач мав доступ до своїх даних, йому потрібно знати спеціальний пароль. Пристрій надає доступ до даних лише після того як пароль буде введено правильно. Він перевіряє чи пароль, що зберігається на пристрої, та пароль, що вводить користувач, є рівними.

Для того, щоб захистити дані у разі втрати пристрою чи іншого витоку даних, записи на карті зберігаються у зашифрованому вигляді. Шифрування відбувається в користувацькій програмі, а пристрій вже приймає зашифровані записи. Для цього використовується AES шифрування [14]. На сервері здійснюється перевірка отриманого OTP. Спочатку сервер розділяє OTP на дві частини: ідентифікатор пристрою та зашифровані дані. Використовуючи ідентифікатор, сервер знаходить відповідний ключ AES-128, що був заздалегідь збережений. Потім сервер розшифровує OTP і перевіряє лічильник використання, дані з внутрішнього таймеру та контрольну суму CRC-16, щоб переконатися в його коректності та унеможливити повторне використання. Якщо всі перевірки успішні, сервер надає доступ до сервісу.

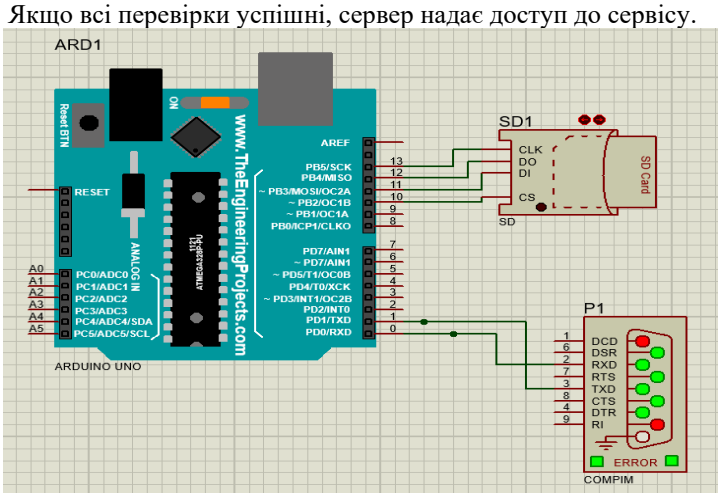

Рис.1. Електрична принципова схема пристрою модельованого модуля

Електрична принципова схема пристрою включає наступні елементи:

- Arduino Uno;
- Virtual SD;
- COMPIM.

SPI використовується для зв'язку між мікроконтролером і картою SD, і він розташований на цифрових контактах 11, 12 і 13 плати Arduino Uno. Для вибору SD-карти також потрібен другий контакт. Можна використовувати контакт SS – 10. Модуль COMPIM під'єднується до Serial портів. У Arduino Uno це контакти 0 та 1.

Програмні рішення

Програма пристрою складається з одного файлу Data_handler.ino. Файл містить наступні функції:

- void setup(); Функція setup виконується першою та налаштовує основні параметри пристрою.
- void loop(); Функція loop пристрій виконує під час роботи та запускає її постійно. В ній визначений алгоритм роботи програми.
- void openFile(FileMode mode); Функція openFile відкриває файл з даними.
- Status addRecord(String recordStr); Функція addRecord додає новий запис у файл.
- Status removeAt(String searchStr); Функція removeAt видаляє запис у файлі.
- Status selectWhere(String searchStr); Функція selectWhere використовується для пошуку запису.

- Status checkAuthorization(String authStr); Функція checkAuthorization перевіряє авторизацію користувача.

Програма для користувача була створена в Visual Studio 2019 за допомогою платформи Windows Forms на мові C#. Windows Forms – це платформа інтерфейсу користувача для створення класичних додатків Windows. Вона забезпечує один з найефективніших способів створення класичних програм за допомогою візуального конструктора в Visual Studio. У Windows Forms можна розробляти графічно складні програми, які легко розгортати, оновлювати, і з якими зручно працювати як в автономному режимі, так і в мережі. Програми Windows Forms можуть отримувати доступ до локального обладнання та файлової системи комп'ютера, на якому працює програма.

Функціональні можливості програми, яка працює за різними сценарієм, який обирає користувач.

Також до функції програми входить шифрування полів запису, перевірка авторизації користувача та створення рядка, що пересилається пристрою.

Прикладна програма для моделювання повинна мати простий та зручний інтерфейс, а також відслідковувати стан виконання операцій, що вибрав користувач (рис.2).

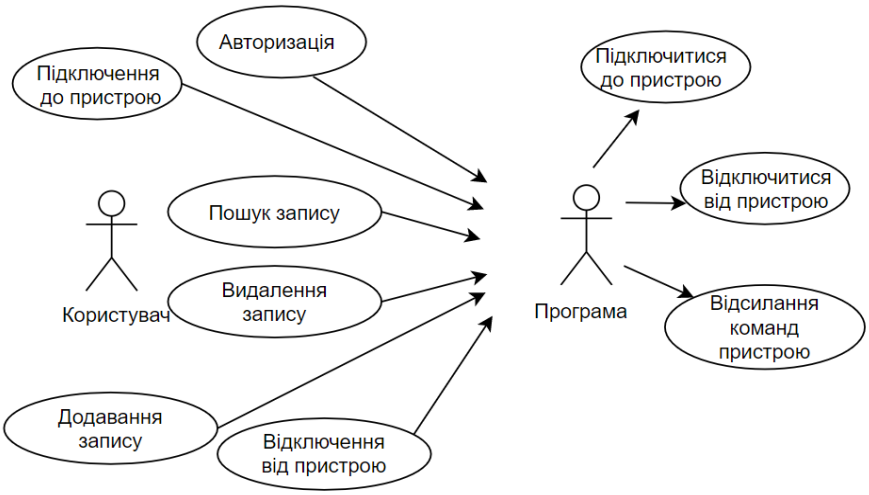

Рис.2. Діаграма прецедентів прикладної програми

Для взаємодії програми з пристроєм використовується спеціальний рядок, що пересилається між ними. Рядки, що відсилаються та приймаються відрізняються та складається з різних частин.

Рядок для відсилання пристрою складається з команди та даних. Для кожної команди поставлено у відповідність певне число.

Рядок, що надходить від пристрою має інший формат та складається з статусу виконуваної операції, даних та слову, що позначає завершення пересилання даних (рис. 3). Статус відповідає стану виконання операції та йому також надається відповідне числове значення.

Рис.3. Командний рядок з повідомленням про завершення приймання даних

Алгоритм, що описує основний функціонал модельованої системи приведено на рисунку 4.

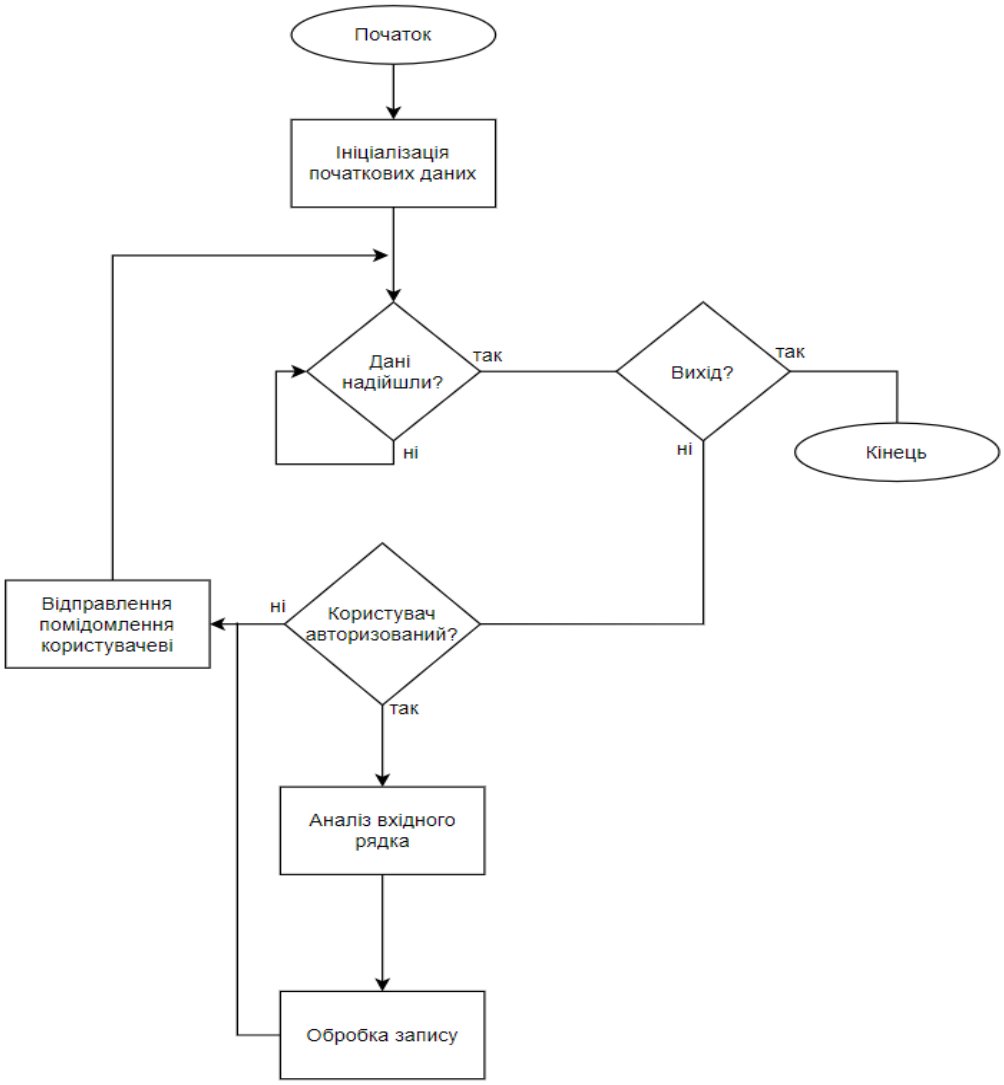

Рис.4. Схема алгоритму роботи програми пристрою

Програма включає в себе ряд файлів, що генеруються платформою Windows Forms [10-11] (рис.5). До основних файлів відносяться Program.cs, Form1.cs та StringChipher.cs.

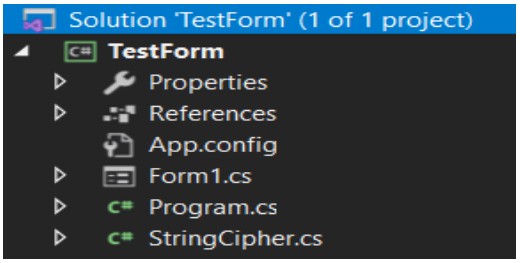

Рис.5. Файли прикладної програми на платформі Windows Forms

Під час старту програми першим виконується файл Program.cs та викликається функція Main(), що запускає початкове вікно.

В файлі Form1.cs міститься клас в якому реалізований основний функціонал програми. Він складається з наступних функцій:

1) form1(); Функція Form1 – це конструктор без параметрів, що виконується першим при створенні об’єкта.
2) void getCOMPorts(object sender, EventArgs e); Функція getCOMPorts виводить всі доступні COM порти.
3) void connectToArduino(); Функція connectToArduino забезпечує з’єднання програми з пристроєм.
4) void disconnectArduino(); Функція disconnectArduino скасовує з’єднання програми з пристроєм.

5) void Form1_FormClosing(object sender, FormClosingEventArgs e); Функція Form1_FormClosing потрібна, для завершення процесів в програмі, наприклад коли користувач забув від'єднатися від пристрою.
6) void sendDataToArduino(object sender, EventArgs e); Функція sendDataToArduino відправляє рядок, що містить дію та дані, пристрою.
7) void connectToArduino(object sender, EventArgs e); Функція connectToArduino з параметрами зв'язує кнопку користувацького інтерфейсу з програмою.
8) void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e); Функція DataReceivedHandler – це подія, яка викликається коли пристрій відправить дані програмі.
9) void rbAdd_Click(object sender, EventArgs e); Функції rbAdd_Click, rbRemove_Click, rbSearch_Click потрібні для правильного перемикання компонентів radioButton.
10) void rbRemove_Click(object sender, EventArgs e);
11) void rbSearch_Click(object sender, EventArgs e);

Також у Form1.cs визначені два перечислення:

- Action.
- Status.

Перечислення Action потрібне для задання дії, що вибрав користувач. Складається з наступних записів:

- ADD_RECORD – команда для добавлення запису.
- REMOVE_RECORD – команда для видалення запису.
- SEARCH_RECORD – команда для пошуку запису.
- AUTHORIZATION – команда для авторизації корисувача.

Перечислення Status містить статуси про виконання операції. Складається з наступних записів:

- OK – статус успішної виконання операції.
- RECEIVED – статус успішного отримання даних.
- SEND – статус відсилання даних пристрою.
- RECORD_MISSING – статус відсутності запису на пристрої .
- ERROR – статус невизначеної помилки.
- ERROR_OPEN_FILE – помилка відкриття файлу на пристрої.
- ERROR_ADD_RECORD – помилки дабавлення нового запису.
- ERROR_DELETE_RECORD – помилка видалення запису.
- ERROR_SEARCH_RECORD – помилку пошуку.
- ERROR_AUTHORIZATION – помилка авторизації.

StringChipher – це клас, що відповідає за шифрування даних AES шифруванням. Містить функції Encrypt та Decrypt для шифрування та розшифрування рядка, а також містить поля з публічним та закритим ключами.

В програмі присутня одна форма, яка створювалася засобами Windows Forms. Програма має наступний вигляд. Для оформлення програми використовувалися стандартні компоненти Windows Forms: кнопки, випадаюче меню, звичайні та розширені текстові поля.

При тестуванні перевірялися основні функції пристрою:

- підключення пристрою до програми користувача;
- авторизація користувача;
- додавання нового запису;
- видалення запису;
- пошук запису.

Для підключенні з випадаючого списку вибирається доступний COM порт та після натискання кнопки "Connect" в компоненті RichTextBox з'явиться запис при успішному з'єднанні.

Для авторизації в поле потрібно ввести секретний ключ. При вдалій авторизації користувач матиме доступ до своїх записів, та зможе взаємодіяти з ними.

Для додавання нового запису потрібно заповнити певні текстові поля та вибрати функцію "Add record".

Після відправки пристрій надасть відповідь про успішність виконання операції, а в файлі, що міститься на образі SD card з'явиться новий запис (рис. 6, 7).

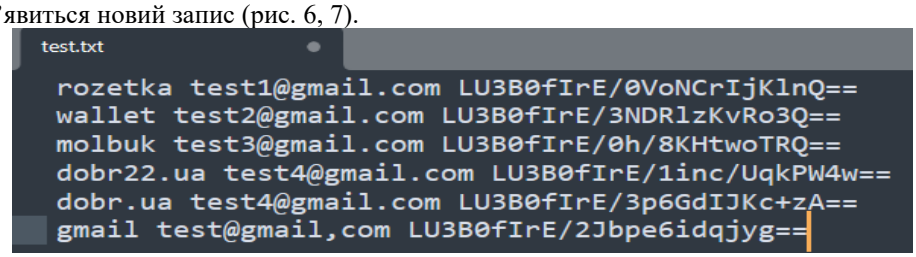

Рис.6. Файли прикладної програми на платформі Windows Forms

Для пошуку потрібно вписати назва запису та вибирається функція "Search record". Після відправки програма буде очікувати на відповідь пристрою. Якщо запис існує програма отримує дані про запис та виводить його в відповідні поля, інакше пристрій надасть відповідь, що запис відсутній.

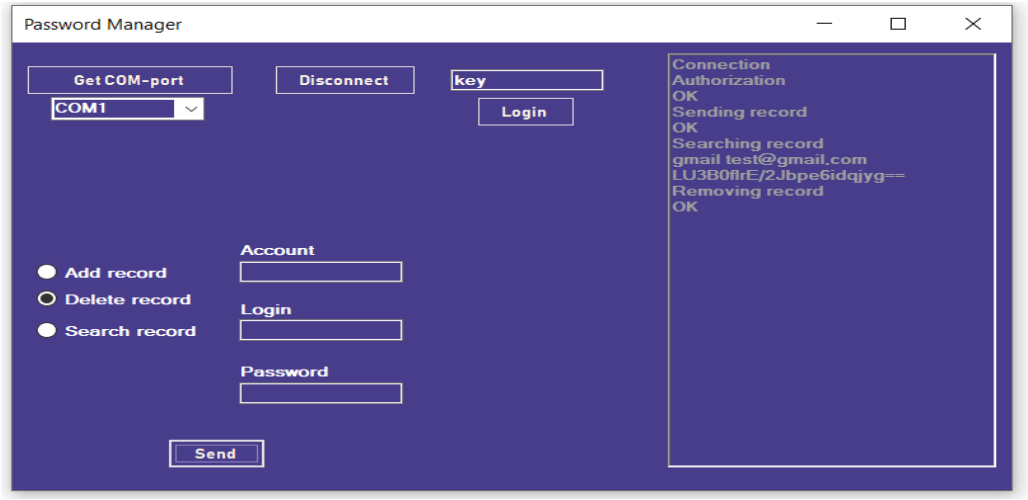

Рис.7. Файли прикладної програми на платформі Windows Forms

Для видалення запису потрібно вписати назва запису та вибрати функція “Delete record”. Після відправки програма буде очікувати на відповідь пристрою. Про вдалість виконанні операції програма отримає певну відповідь. Початок запису, що міститься у файл на образі SD card, буде позначений символом “|”. Програма на пристрої буде рахувати що цей запис видалений та пропустить його.

Висновки

В результаті виконання досліджень проведено аналіз і класифікацію способів зберігання персоніфікованих цифрових даних та типів пристроїв для збереження, зокрема розглянута їх архітектура та особливості. Запропоновано архітектуру і розглянуто загальну модель пристрою, зберігання даних, а т для багаторівневого доступу до конфіденційних даних, обгрунтовано формат даних та їх захист.

Проведене моделювання дозволяє підтвердити можливість вирішення кількох важливих завдань: аналізувати вимоги до захисту інформації за критеріями конфіденційність, доступність цілісність; в залежності від критерію «важливість» та «своєчасність» сортувати дані та визначати рівень доступу до них і тривалість зберігання, що дозволяє оптимізувати апаратні ресурси системи та накопичення персональної інформації на них.

Створена прикладна програма для користувача є демонстраційною. Для подальших досліджень доцільно опрацювати підходи з автоматизованого сортування персональної інформації за вказаними вище критеріями. Корисним був би підхід, який дозволить реалізувати проаналізовані методи засобами програмно і апаратно реконфігурованих ресурсів.

Література

1. Комплексна безпека інформаційних мережевих систем: навчальний посібник для студентів спеціальності 174 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» / Укладачі: А.Г. Микитишин, М.М. Митник, О.С. Голотенко, В.В. Карташов. – Тернопіль : ФОП Паляниця В.А., 2023. – 324 с.
2. Рудий Р.; Воробець Г. Апаратні і програмні рішення захисту персоніфікованих даних та інформації в телекомунікаційних системах і мережах. // Herald of Khmelnytskyi National University. Technical sciences 2025, 357 (5.2), 48-54. https://doi.org/10.31891/2307-5732-2025-357-65.
3. Рябчинська О.П., Стоматов Е.Г. Конфіденційна інформація про особу як предмет кримінального правопорушення, передбаченого ст. 182 КК України: змістовні ознаки. // Юридичний науковий електронний журнал. – № 6/2023. – С.484-490. DOI https://doi.org/10.32782/2524-0374/2023-6/113
4. Буртник Х. Конфіденційна інформація, інформація про особу та персональні дані: співвідношення і регулювання. – https://cedem.org.ua/analytics/konfidentsijna-informatsiya-informatsiya-pro-osobu-ta-personalni-dani-spivvidnoshennya-i-regulyuvannya/
5. Мінцер О. П., Горшков Є. В., Твердохліб О. І. та ін. Персоніфікована медицина як глобальна об’єднуюча мета розвитку страхової та системної медицини. // Медична інформатика та інженерія. – 2019, № 2. – С.38-44. DOI: https://doi.org/10.11603/mie 1996-1960.2019.2.10317
6. Damjanovski V. CCTV Networking and Digital Technology / Vlado Damjanovski. – Sydney: Elsevier, 2013. – 614 с.
7. Tate J. Introduction to Storage Area Networks / J. Tate, P. Beck, L. Miklas: RedBooks, 2017. – 280 с.
8. NAS vs SAN. URL: https://hypertecdirect.com/knowledge-base/nas-vs-san/.
9. What is data storage? URL: https://www.ibm.com/topics/data-storage
10. Memory. URL: https://www.arduino.cc/en/Tutorial/Foundations/Memory
11. Overview of the Arduino UNO Components. URL: https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board
12. Bayle J. C Programming for Arduino / Julien Bayle. – Birmingham: Packt Publishing, 2013. – 488 с.

13. SD. URL: https://www.arduino.cc/reference/en/libraries/sd/
14. Paar C. Understanding Cryptography / C. Paar, J. Pelzl. – Bochum: Springer, 2013. – 372 c.

References

1. Kompleksna bezpeka informatsiinykh merezhevykh system: navchalnyi posibnyk dlia studentiv spetsialnosti 174 «Avtomatyzatsiia, kompiuterno-intehrovani tekhnolohii ta robototekhnika» / Ukladachi: A.H. Mykytyshyn, M.M. Mytnyk, O.S. Holotenko, V.V. Kartashov. – Ternopil : FOP Palianytsia V.A., 2023. – 324 s.
2. Rudyi R.; Vorobets H. Aparatni i prohramni rishennia zakhystu personifikovanykh danykh ta informatsii v telekomunikatsiinykh systemakh i merezhakh. // Herald of Khmelnytskyi National University. Technical sciences 2025, 357 (5.2), 48-54. https://doi.org/10.31891/2307-5732-2025-357-65.
3. Riabchynska O.P., Stomatov E.H. Konfidentsiina informatsiia pro osobu yak predmet kryminalnoho pravoporushennia, peredbachenoho st. 182 KK Ukrainy: zmistovni oznaky. // Yurydychnyi naukovyi elektronnyi zhurnal. – № 6/2023. – S.484-490. DOI https://doi.org/10.32782/2524-0374/2023-6/113
4. Burtnyk Kh. Konfidentsiina informatsiia, informatsiia pro osobu ta personalni dani: spivvidnoshennia i rehuliuvannia. – https://cedem.org.ua/analytics/konfidentsijna-informatsiya-informatsiya-pro-osobu-ta-personalni-dani-spivvidnoshennya-i-regulyuvannya/
5. Mintser O. P., Horshkov Ye. V., Tverdokhlib O. I. ta in. Personifikovana medytsyna yak hlobalna obiednuiucha meta rozvytku strakhovoi ta systemnoi medytsyny. // Medychna informatyka ta inzheneriia. – 2019, № 2. – S.38-44. DOI: https://doi.org/10.11603/mie 1996-1960.2019.2.10317
6. Damjanovski V. CCTV Networking and Digital Technology / Vlado Damjanovski. – Sydney: Elsevier, 2013. – 614 s.
7. Tate J. Introduction to Storage Area Networks / J. Tate, P. Beck, L. Miklas: RedBooks, 2017. – 280 s.
8. NAS vs SAN. URL: https://hypertecdirect.com/knowledge-base/nas-vs-san/.
9. What is data storage? URL: https://www.ibm.com/topics/data-storage
10. Memory. URL: https://www.arduino.cc/en/Tutorial/Foundations/Memory
11. Overview of the Arduino UNO Components. URL: https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board
12. Bayle J. C Programming for Arduino / Julien Bayle. – Birmingham: Packt Publishing, 2013. – 488 s.
13. SD. URL: https://www.arduino.cc/reference/en/libraries/sd/
14. Paar C. Understanding Cryptography / C. Paar, J. Pelzl. – Bochum: Springer, 2013. – 372 s.