МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО – КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кваліфікаційна наукова
праця на правах рукопису

ДАНИЛЬЧЕНКО ВАЛЕНТИНА МИКОЛАЇВНА

УДК 004.382.7

ДИСЕРТАЦІЯ

МЕТОДИ ОБРОБКИ ДАНИХ В МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСАХ НА ОСНОВІ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ

Спеціальність 123 «Комп'ютерна інженерія»
Галузь знань 12 «Інформаційні технології»
Подається на здобуття наукового ступеня
доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело_____В.М. Данильченко

Науковий керівник: СРІБНА Ірина Миколаївна, доктор технічних наук, доцент

Електронний підпис
Данильченко Валентина Миколаївна
Ідентифікаційний код 3422101242
Підписано у Вчасно

Київ – 2023

АНОТАЦІЯ

Данильченко В.М. Методи обробки даних в мікрокомп'ютерних комплексах на основі персональної ідентифікації. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 – комп'ютерна інженерія (галузь знань 12 – інформаційні технології). – Державний університет інформаційно-комунікаційних технологій, Міністерства освіти і науки України, Київ, 2023.

Дисертаційна робота присвячена актуальній науковій задачі розробки методів обробки даних в мікрокомп'ютерних комплексах на основі персональної ідентифікації.

Ефективна обробка даних є ключовим етапом у дослідженнях та різноманітних галузях, оскільки вона впливає на точність, достовірність та узагальнення отриманих результатів. Обробка даних усуває аномалії, виявляє та коригує помилки, що можуть виникнути під час збору або введення даних. Це допомагає забезпечити точність аналізу та результатів дослідження.

Для забезпечення високих вимог до кількісних і якісних показників обробки інформації, що є процесом інформаційної взаємодії технологічних процесів сучасних підприємств і організацій, необхідно вдосконалення алгоритмів обробки даних, які дозволяють покращити точність та надійність систем персональної ідентифікації.

Для досягнення цього завдання необхідно розробити програмне забезпечення для проведення персональної ідентифікації, що дозволить проводити оцінку ефективності завадостійкого коду, витрат пам'яті та ступеня захищеності даних, що в свою чергу забезпечить надійність мікрокомп'ютерних комплексів.

У роботі проведено аналіз особливостей функціонування та організації зберігання інформації в персональних ідентифікаторах МКПІ (мікрокомп'ютерних комплексів персональної ідентифікації), а також

можливостей підвищення надійності шляхом модифікації обробки, зберігання та аналізу помилок, характерних для МКПІ.

Міжнародні стандарти обмежують розміри чіпа для забезпечення сумісності з будь-якими комп'ютерами. Виробники персональних ідентифікаторів не гарантують надійності та довговічності зберігання інформації в них, а самі пристрої часто не мають механізмів забезпечення надійності зберігання даних та стійкості до перешкод та помилок. У свою чергу розвиток елементної бази та технологічних процесів виготовлення напівпровідникових виробів сприяє збільшенню пам'яті персональних ідентифікаторів та зростанню продуктивності.

У дисертаційній роботі розроблено методи та алгоритми обробки та зберігання інформації, адаптовані для представників класу МКПІ, які будуть стійкі до виникнення помилок у пам'яті персональних ідентифікаторів, обумовлених різними випадковими процесами, та спроектовано схеми завадостійкого кодування. Надані рекомендації щодо підвищення надійності МКПІ. Запропоновано зміни до базових стандартів для SMART-карт, магнітних карт, біометричних паспортів та RFID-систем, що регулюють обробку та зберігання даних.

Мета дисертації – полягає у розробці та впроваджені методів обробки даних в мікрокомп’ютерних комплексах на основі персональної ідентифікації.

Спотворення інформації можуть виникати внаслідок випадкових процесів, таких як шум або помилки в передачі даних. Нові методи кодування інформації повинні бути здатними виявляти та виправляти такі помилки, щоб відновити початкову інформацію.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Проаналізувати інформацію про наявні мікрокомп'ютерні системи, які використовуються для особистої ідентифікації.

2. Дослідити вплив математичних моделей випадкового та послідовного вибору сторінок для обробки даних на ефективність завадостійкого коду, витрати пам'яті та ступінь захищеності даних.

3. Розробити алгоритми, які дозволяють ефективно обробляти біометричні, текстові чи інші дані для ідентифікації особи в мікрокомп'ютерних системах.

4. Дослідити методи обробки, зберігання та введення-виведення даних в мікрокомп'ютерних комплексах персональної ідентифікації на основі технології RFID.

5. Розробити метод обробки та введення-виведення даних для підвищення надійності та захищеності даних.

Методи дослідження. Результати проведених і представлених в дисертації досліджень отримані з використанням методів теорії ймовірностей та математичної статистики, дискретної математики, математичного аналізу та шляхом експериментальних досліджень.

Наукова новизна дисертації:

1. Розроблено модель для випадкового та послідовного вибору сторінок для введення-виведення даних, що дозволяє оцінювати алгоритми завадостійкого кодування інформації в МКПІ для розрахунку ефективності завадостійкого коду.
2. Набув подальшого розвитку метод обробки, зберігання та ведення-виведення даних у МКПІ, що дозволяє знизити енерговитрати у RFID пристроях.
3. Розроблено метод обробки та введення-виведення даних, що дозволяє підвищити надійність та захищеність даних на основі синхронізації даних.

Практична цінність роботи підтверджується використанням результатів роботи у системі контролю доступу на основі технології RFID у ТОВ «Хуавей Україна». Розроблені рішення використовуються для виконання практичних завдань, пов'язаних з підвищенням надійності систем

радіочастотної ідентифікації. Відповідно до вимог, пред'явлених до системи RFID-ідентифікації, методи кодування та декодування даних були повністю перенесені на хост-комп'ютер. Випробування показали корекцію помилок у 89% випадків.

Застосування результатів роботи. Результати роботи використані у ТОВ «Хуавей Україна» для вирішення практичних завдань пов'язаних із підвищенням надійності персональних пристроїв на базі SMART-карток, що підтверджують повноваження доступу до спеціалізованих баз даних. Результати досліджень дозволили на 23% підвищити надійність SMART-карток з персональними відомостями. У 89% випадків дані в МКПІ було кориговано.

Дисертаційна робота виконана відповідно до планів наукової і науково-технічної діяльності Державного університету інформаційно-комунікаційних технологій і є частиною досліджень в рамках науково-дослідної роботи «Дослідження обробки траєкторної інформації в вимірювально-обчислювальних системах» (ДУІКТ, м. Київ), а також госпдоговірної роботи «Розробка системи активного управління чергою пакетів в мережах TCP/IP з використанням REM-регуляторів» (№ 0119U101284, ДУТ, м. Київ).

Ключові слова: мікрокомп’ютерні комплекси, код Хеммінга, код Ріда-Соломона, RFID, SMART–карта, магнітні картки, база даних, персональна ідентифікація, надійність, комп’ютерна технологія штрих-коду, IoT.

ANNOTATION

Danylchenko V.M. Methods of data processing in microcomputer complexes of personal identification. - Qualifying scientific work on the rights of the manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 123 - computer engineering (field of knowledge 12 - information technology). - State University of information and communication technologies of the Ministry of Education and Science of Ukraine, Kyiv, 2023.

The dissertation is devoted to the current scientific problem of developing data processing methods in microcomputer complexes based on personal identification. The topic of the dissertation research corresponds to the standard and professional competences of the educational and scientific program for training doctors of philosophy in computer engineering of the State University of Information and Communication Technologies of the Ministry of Education and Science of Ukraine, namely: fundamental scientific research of theoretical-methodological, scientific-methodological and applied principles of efficiency improvement innovative and production activity of the enterprise, as well as improving the process of ensuring the introduction of the latest information technologies at the objects of information activity.

Effective data processing is a key step in research and a variety of industries, as it affects the accuracy, reliability, and generalizability of the results obtained. Data processing eliminates anomalies, detects and corrects errors that may occur during data collection or input. This helps ensure the accuracy of the analysis and research results.

To ensure high requirements for quantitative and qualitative indicators of information processing, which is a process of information interaction of technological processes of modern enterprises and organizations, it is necessary to improve data processing algorithms that allow improving the accuracy and reliability of personal identification systems.

To achieve this task, it is necessary to develop software for conducting personal identification, which will allow to evaluate the effectiveness of the interference-resistant code, memory consumption and the degree of data security, which in turn will ensure the reliability of microcomputer complexes.

The paper analyzes the features of the functioning and organization of information storage in personal identifiers of the MKPI (microcomputer complexes of personal identification), as well as the possibilities of increasing reliability by modifying the processing, storage and analysis of errors characteristic of the MKPI.

International standards limit the size of the chip to ensure compatibility with any computer. Manufacturers of personal identifiers do not guarantee the reliability and durability of information storage in them, and the devices themselves often do not have mechanisms to ensure the reliability of data storage and resistance to interference and errors. In turn, the development of the element base and technological processes of manufacturing semiconductor products contributes to the increase in the memory of personal identifiers and the growth of productivity.

In the dissertation work, methods and algorithms of information processing and storage, adapted for representatives of the MKPI class, which will be resistant to the occurrence of errors in the memory of personal identifiers due to various random processes, have been developed, and interference-resistant coding schemes have been designed. Recommendations are given to improve the reliability of the MKPI. Changes to the basic standards for SMART cards, magnetic cards, biometric passports and RFID systems regulating data processing and storage are proposed.

The purpose of the dissertation is to develop and implement data processing methods in microcomputer systems based on personal identification.

Information distortions can occur due to random processes, such as noise or errors in data transmission. New methods of encoding information must be able to detect and correct such errors in order to restore the original information.

To achieve the goal, the following tasks must be solved:

1. Analyze information about existing microcomputer systems used for personal identification.

2. Investigate the impact of mathematical models of random and sequential page selection for data processing on the effectiveness of the tamper-resistant code, memory consumption, and the degree of data security.

3. To develop algorithms that allow efficient processing of biometric, text or other data for personal identification in microcomputer systems.

4. To study methods of data processing, storage and input-output in microcomputer complexes of personal identification based on RFID technology.

5. Develop a method of data processing and input-output to increase data reliability and security.

Research methods. The results of the research conducted and presented in the dissertation were obtained using the methods of probability theory and mathematical statistics, discrete mathematics, mathematical analysis and through experimental research.

Scientific novelty of the dissertation:

1. A model has been developed for the random and sequential selection of pages for data input-output, which allows to evaluate the algorithms of interference-resistant coding of information in the ICPI to calculate the effectiveness of the interference-resistant code.

2. The method of processing, storing and maintaining and outputting data in the ICPI has gained further development, which allows reducing energy consumption in RFID devices.

3. A method of data processing and input-output has been developed, which makes it possible to increase the reliability and security of data based on data synchronization.

The practical value of the work is confirmed by the use of the results of the work in the access control system based on RFID technology at Huawei Ukraine LLC. The developed solutions are used to perform practical tasks related to increasing the reliability of radio frequency identification systems. In accordance

with the requirements for the RFID identification system, the data encoding and decoding methods have been completely transferred to the host computer. Tests showed error correction in 89% of cases.

Application of work results. The results of the work were used in "Huawei Ukraine" LLC to solve practical tasks related to increasing the reliability of personal devices based on SMART cards, which confirm the authorization of access to specialized databases. The research results made it possible to increase the reliability of SMART cards with personal information by 23%. In 89% of cases, the data in the ICPI was corrected.

The dissertation work was carried out in accordance with the plans of scientific and scientific and technical activities of the State University of Information and Communication Technologies and is part of the research within the framework of the research work "Investigation of the processing of trajectory information in measuring and computing systems" (DUIKT, Kyiv), as well as the contract work "Development of a system of active packet queue management in TSR/IR networks using REM-regulators" (No. 0119U101284, DUT, Kyiv).

Key words: microcomputer complexes, Hamming code, Reed-Solomon code, RFID, SMART card, magnetic cards, database, personal identification, reliability, barcode computer technology, IoT.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. О. М. Ткаленко, В.М. Чорна, *“* Аналіз застосування цифрового нечіткого регулятора в системі АРПП при дії адитивних завад*”, Науково-технічний журнал “Сучасний захист інформації”*, №2, с. 42- 46, 2018.
2. В.М. Чорна, О. М. Ткаленко, О. В. Полоневич, О. В. Сеньков, *“* Особливості захисту інформації в NFC.*”, Науковий журнал “Зв’язок”,* №4, с. 25-29, 2018.
3. К. П. Сторчак, А. М. Тушич, О.М. Ткаленко, Т.М. Жила, В.М. Чорна, “ Аналіз вимог до проектування хмарної платформи для Інтернету речей.” *Науковий журнал “Зв’язок”, №4,* с. 30-35, 2019.
4. О. А. Золотухіна, О. М. Ткаленко, А. М. Тушич, В.М. Чорна, О. Р. Нікітенко, *“* Концепція розвитку підсистеми передавання мультимедійних повідомлень IMS*”, Науково-технічний журнал “Телекомунікаційні та інформаційні технології”*, №4, с. 81- 89, 2019.
5. Vitalii Savchenko, Volodymir Druzhynin, Mykola Tverdohlib, Yevhen Ivanichenko, Nadiia Dovzhenko, Zoreslava Brzhevska, Valentina Chorna, “Analysis and Design of a Hybrid Load Management method for the IoT Networks”, *International Journal of Advanced Trends in Computer Science and Engineering.* vol. 9, no 1, pp. 552-557, 2020.
6. Danylchenko V.M., Mykolaichuk V.R., Tkalenko O.M., Ditkivskyi A.S “Initial setup of pbx server based on Asterisk”, *Науковий журнал “Зв’язок”,* №6, с. 49-53, 2020.
7. В.М. Данильченко, І.М. Срібна, “Розроблення комп'ютерного детектора штрих-коду на основі одного чіпа мікрокомп'ютера”, *Науковий журнал “Зв’язок”,* №3, с. 51-53, 2021.
8. В.М. Данильченко, “ Дослідження мобільної онлайн системи блокування мікрокомп’ютера на основі технології NFC”, *Науковий журнал “Зв’язок”,* №5, с. 20-24, 2023.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

9. О.М. Ткаленко, В.М. Чорна, *“*Вимоги до сучасних телекомунікаційних систем*”* на *VII Науково-технічній конференції*

студентів та молодих вчених факультету Інформаційних технологій "Сучасні інфокомунікаційні технології". - К.ДУТ, 2018 – 5 грудня 2018 р. – С.14-15.

10. В.М. Чорна, Є.Є. Белінський, "Перспективна архітектура стандарту 802.11" на *VI Науково-технічній конференції студентів та молодих вчених факультету Інформаційних технологій "Сучасні інфокомунікаційні технології"*- К.ДУТ, 2018 - 25 травня 2018р. – С.19-20.

11. В.М. Чорна, С. А. Жежкун, " Системи оптимізації енергоспоживання для розумного будинку", на *XIII міжнар. науко-во-технічної конференції «Проблеми інформатизації». Київ: ДУТ, НТУ; Полтава: ПНТУ; Катовице: КЕУ; Париж: Університет Париж VII Вєнсент-Сен-Дені; Вільнюс: ВДТУ; Харків : ХНДІТМ: Білорусь: БДАЗ; Кропивницький:* ЛА НАУ, 2019. – 172 с.

12. І.М. Срібна, В.М. Чорна, "Міжмережевий екран – один з механізмів захисту мережі" на *IX Науково-технічної конференції студентів та молодих вчених Навчально-наукового інституту Інформаційних технологій "Сучасні інфокомунікаційні технології"*- К.ДУТ, 2019 – 5 грудня 2019р. – С.190-192.

13. В.М. Чорна, С.М. Брезіцький "Прогнози впровадження Інтернету речей" на *XIII міжнар. науко-во-технічної конференції «Проблеми інформатизації». Київ: ДУТ, НТУ; Полтава: ПНТУ; Катовице: КЕУ; Париж: Університет Париж VII Вєнсент-Сен-Дені; Вільнюс: ВДТУ; Хар-ків : ХНДІТМ: Білорусь: БДАЗ; Кропивницький:* ЛА НАУ, 2019. – 120 с.

14. В.Р. Миколайчук, І.М. Срібна, В.М. Чорна, "Вибір програмного забезпечення для передавання голосу по мережам з маршрутизацією пакетів", на *Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаціних технологіях»*. Київ, 2020 року – 35-37 с.

15. В.М. Чорна, В.М. Постельников "Дослідження методів обробки великих масивів данних в системах IoT" на *Всеукраїнській науково-технічній конференції «Сучасний стан та перспективи розвитку IoT»*. Київ, 2020. – 156с.

16. Danylchenko V.M. Mykolaichuk V.R., "Improving the reliability of storing information in microcomputer complexes of personal identification", на *XI міжнародна науково-технічної конференції студентства «Світ інформації та телекомунікації»*. Київ, 2021 року – 218-220 с.

ЗМІСТ

ВСТУП…….. 15
РОЗДІЛ 1. АНАЛІЗ МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСІВ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ, ЗАПИС, ЗБЕРІГАННЯ І ВВЕДЕННЯ-ВИВЕДЕННЯ ДАНИХ ..…….. 21
1.1 Мікрокомп'ютерний комплекси персональної ідентифікації, зберігання даних і особливості кодування інформації ……. 21
1.1.1 Мікрокомп’ютерні комплекси персональної ідентифікації на осново технології RFID ……. 40
1.1.2 Біометричні паспорти як особливий персональний ідентифікатор ……... 49
1.1.3 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації ……. 54
1.2 Аналіз помилок, виникаючих в мікрокомп'ютерних комплексах персональної ідентифікації …… 57
1.2.1 Можливості застосування підвищення надійності SMART – карт ……. 57
1.2.2 Можливості підвищення надійності магнітних карт …… 62
1.2.3 Можливості підвищення надійності в мікрокомп’ютерних комплексах на основі технології RFID ……. 63
1.2.4 Можливості підвищення надійності біометричних паспортів ……... 64
1.3 Аналіз помилок, виникаючих в мікрокомп’ютерних комплексах персональної ідентифікації ……. 65
1.3.1 Помилки, характерні для магнітних карт …… 66
1.3.2 Помилки, характерні для мікрокомп’ютерних комплексів на основі технології RFID ……... 66
1.3.3 Помилки, характерні для біометричних паспортів ……. 67
РОЗДІЛ 2. РОЗРОБКА МЕТОДІВ І АЛГОРИТМІВ ОБРОБКИ, ЗБЕРІГАННЯ ТА ВВЕДЕННЯ-ВИВЕДЕННЯ ДАНИХ В МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСАХ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ …… 70
2.1 Схеми організації введення-виведення в комплексах персональної ідентифікації …….. 70
2.2 Розробка методів і алгоритмів обробки, зберігання та введення-виведення даних в SMART-картах …….. 75
2.2.1 Модель випадкового вибору сторінок пам'яті для введення-виведення даних ……..... 77

2.2.2 Застосування коду Хеммінга для обробки та зберігання інформації у SMART-картах ……... 84
2.2.3 Застосування коду Ріда-Соломона для обробки та зберігання інформації у SMART-картах …… 103
2.2.4 Застосування локального коду надлишковості для обробки та зберігання інформації в SMART-картах …… 111
2.3 Розробка методів і алгоритмів обробки, зберігання та введення-виведення даних в мікрокомп'ютерних комплексах персональної ідентифікації на основі технології RFID ……. 112
2.4 Розробка методів і алгоритмів обробки, зберігання та введення-виведення даних в біометричних паспортах …… 115
2.4.1 Застосування коду Хеммінга для обробки та зберігання інформації в біометричних паспортах ……... 117
2.4.2 Застосування коду Ріда-Соломона для обробки та зберігання інформації у біометричних паспортах ……. 118
РОЗДІЛ 3 МОДИФІКАЦІЯ ТА ЗАСТОСУВАННЯ МЕТОДІВ ЗБЕРІГАННЯ ТА ВВЕДЕННЯ-ВИВЕДЕННЯ ДАНИХ В МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСАХ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ …… 120
3.1 Модифікація методів зберігання та введення-виведення даних в SMART-картах ……..... 120
3.1.1 Модифікація файлової системи SMART-карти ……... 120
3.1.2 Реалізація введення-виведення даних у SMART-картах ….. 128
3.2 Модифікація методів зберігання даних у магнітних картах …… 136
3.3 Модифікація методів зберігання та введення-виведення даних в мікрокомп'ютерний комплексах персональної ідентифікації на основі технології RFID ……. 139
3.4 Модифікація методів зберігання та введення-виведення даних у біометричних паспортах ……..... 142
3.4.1 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації на базі SMART-карт ……. 147
3.4.2 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації на базі магнітних карток …… 152
3.4.3 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації на основі технології RFID ……... 154
3.5 Розробка системи динамічного RFID ключа в мікрокомп'ютерних комплексах …….. 156
ВИСНОВКИ …… 161

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ …….. 164
ДОДАТОК А ……... 178
ДОДАТОК Б …… 184
ДОДАТОК В …… 186
ДОДАТОК Г …… 188

ВСТУП

Актуальність теми.

Чи можливо уявити справжній етап розвитку інформаційного суспільства без таких результатів науки і техніки, як SIM-карта мобільного телефону, карта оплати проїзду на транспорті, банківська пластикова карта, біометричний паспорт. Це лише частина технічних рішень, яка є в постійному користуванні практично у кожного. Список подібних пристроїв можна довго перераховувати, з кожним роком з'являються нові зразки, більш портативні, більш продуктивні. Сфери виробництва, послуг, логістики, безпеки, медицини володіють власними вузькопрофільними зразками, які виконують прикладні функції: електронні ключі доступу (e-token), картки лояльності, чіп-ключі автомобіля, паркувальні автомати, цифрове телебачення. Всі вони характеризуються тим, що надовго вписалися в наше життя і з різним ступенем наближеності організовують життєдіяльність кожного.

Особливості архітектури та функціонального призначення даних систем дозволяють дати їм назву *Мікрокомп'ютерні комплекси персональної ідентифікації* (МКПІ). В даний час МКПІ реалізують вкрай важливу функцію, вони є носіями виняткових даних власника. З огляду на найбільшій (в порівнянні з усіма іншими автоматизованими системами) наближеності до різних суспільних процесів і безпосередньо до людини (власнику) цінність інформації в таких комплексах набагато перевищують їх ринкову вартість. Втрата або спотворення інформації є критичними і не допустимі для них. Таким чином, довготривале зберігання і своєчасне надання інформації є ключовою функцією кожного персонального ідентифікатора. У відкритій пресі практично відсутні відомості про шляхи модернізації та вдосконаленні МКПІ в частині підвищення надійності за допомогою розробки і модифікації методів і алгоритмів обробки, зберігання та обробки інформації. Дані пристрої часто не використовують алгоритмів забезпечення цілісності

інформації та надійності функціонування, що може породити виникнення помилок, багато в чому, що не залежать від дій власника. Одна з проблем мікро-чіпів - це виникнення можливої несправності в сторінково-орієнтованих модулях пам'яті. Незахищеність магнітної смуги, в разі її пошкодження може привести до незворотної втрати даних і заміні карти. Вплив ненавмисних (паразитних) і навмисних електромагнітних зв'язків і перешкод, наявність яких впливає на ефективність роботи. Виникнення випадкових помилок запису, модифікації даних, що зберігаються, можуть не зворотно вплинути на працездатність пристрою. Тому, з плином часу, зростає ймовірність виникнення помилок в МКПІ. Існують недоліки в підходах до обробки та зберігання даних, що забезпечують підвищення надійності персональних ідентифікаторів та достовірність виконання ключових функцій МКПІ. Питання підвищення надійності функціонування МКПІ і поліпшення їх техніко-економічних характеристик шляхом використання алгоритмів спеціального кодування в даний час в науковій літературі практично не розглянуті. Відомі алгоритми орієнтовані на системи високої продуктивності і не розраховані на використання в пристроях з обмеженими обсягами пам'яті і невеликими обчислювальними ресурсами.

У зв'язку з цим, метою дослідження є розробити нові методи кодування інформації для МКПІ, які будуть більш ефективніше захищати інформацію від спотворень і помилок. Спотворення інформації можуть виникати внаслідок випадкових процесів, таких як шум або помилки в передачі даних. Нові методи кодування інформації повинні бути здатні виявляти та виправляти такі помилки, щоб відновити початкову інформацію.

Зв’язок роботи з науковими програмами, планами, темами.

Тематика дисертаційної роботи і отримані результати безпосередньо відповідають пріоритетності розвитку інформаційних та комунікаційних технологій в Україні до 2021 р. згідно із Законом України «Про пріоритетні напрями розвитку науки і техніки», від 11.07.2001 № 2623-III, зі змінами

внесеними згідно із Законом України «Про наукову та науково-технічну діяльність» від 26.11.2015 № 848-VIII. Дисертаційна робота виконана відповідно до планів наукової і науково-технічної діяльності Державного університету інформаційно-комунікаційних технологій і є частиною досліджень в рамках науково-дослідної роботи «Дослідження обробки траєкторної інформації в вимірювально-обчислювальних системах» (ДУТ, м. Київ), а також госпдоговірної роботи «Розробка системи активного управління чергою пакетів в мережах TCP/IP з використанням REM-регуляторів» (№ 0119U101284, ДУТ, м. Київ).

Мета і завдання дослідження.

Мета дисертації - полягає у розробці та впроваджені методів обробки даних в мікрокомп’ютерних комплексах на основі персональної ідентифікації.

Для досягнення цієї мети необхідно вирішити такі *задачі*:

1. Проаналізувати інформацію про наявні мікрокомп'ютерні системи, які використовуються для особистої ідентифікації.

2. Дослідити вплив математичних моделей випадкового та послідовного вибору сторінок для обробки даних на ефективність завадостійкого коду, витрати пам'яті та ступінь захищеності даних.

3. Розробити алгоритми, які дозволяють ефективно обробляти біометричні, текстові чи інші дані для ідентифікації особи в мікрокомп'ютерних системах.

4. Дослідити методи обробки, зберігання та введення-виведення даних в мікрокомп'ютерних комплексах персональної ідентифікації на основі технології RFID.

5. Розробити метод обробки та введення-виведення даних для підвищення надійності та захищеності даних.

Об'єктом дослідження є процес функціонування мікрокомп’ютерних комплексів для персональної ідентифікації.

Предметом дослідження є методи і алгоритми, зберігання та обробки даних, спрямовані на підвищення надійності за допомогою впровадження кодів корекції помилок в мікропроцесорні програми МКПІ.

Методи дослідження. Як теоретичну та методологічну основу дисертаційного дослідження використано елементи теорії ймовірностей та математичної статистики, дискретної математики, математичного аналізу та шляхом експериментальних досліджень.

Наукова новизна одержаних результатів полягає у наступному:

1. Розроблено модель для випадкового та послідовного вибору сторінок для введення-виведення даних, що дозволяє оцінювати алгоритми завадостійкого кодування інформації в МКПІ для розрахунку ефективності завадостійкого коду.

2. Набув подальшого розвитку метод обробки, зберігання та ведення-виведення даних у МКПІ, що дозволяє знизити енерговитрати у RFID пристроях.

3. Розроблено метод обробки та введення-виведення даних, що дозволяє підвищити надійність та захищеність даних на основі синхронізації даних.

Практичне значення отриманих результатів.

Практична цінність роботи підтверджується використанням результатів роботи у системі контролю доступу на основі технології RFID у ТОВ «Хуавей Україна». Розроблені рішення використовуються для виконання практичних завдань, пов'язаних з підвищенням надійності систем радіочастотної ідентифікації. Відповідно до вимог, пред'явлених до системи RFID-ідентифікації, методи кодування та декодування даних були повністю перенесені на хост-комп'ютер. Випробування показали корекцію помилок у 89% випадків.

Особистий внесок здобувача. Всі положення, які виносяться на захист, належать особисто автору. В роботах, які опубліковано в співавторстві, особисто здобувачу належать: проаналізовані вимоги до проектування

хмарної платформи на основі бездротової мережі [4], досліджено концепцію розвитку підсистеми передавання мультимедійних повідомлень з використанням бездротових мереж [3], досліджено методи організації прав користувачів у системах [3], дослідження застосування методів і алгоритмів обробки та зберігання даних в мікрокомп'ютерних комплексах персональної ідентифікації [1], досліджені сучасні телекомунікаційні системи [2], розроблені методи зберігання та обробки даних в мікрокомп’ютерних комплексах персональної ідентифікації. [5]

Апробація результатів дисертації. Основні результати дисертаційних досліджень доповідалися й обговорювалися на таких конференціях:

- XI Міжнародна науково-технічна конференції студентства «Світ інформації та телекомунікації». (Київ, 2021 року);
- Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаціних технологіях». (Київ, 2020 року);
- Всеукраїнській науково-технічній конференції «Сучасний стан та перспективи розвитку IoT». (Київ, 2020 року);
- XIII Міжнародна науково-технічна конференції «Проблеми інформатизації». (Київ: ДУТ, НТУ; Полтава: ПНТУ; Катовице: КЕУ; Париж: Університет Париж VII Вєнсент-Сен-Дені; Вільнюс: ВДТУ; Харків, ХНДІТМ: Білорусь: БДАЗ; Кропивницький: ЛА НАУ, 2019 року);
- IX Науково-технічна конференція студентів та молодих вчених Навчально-наукового інституту Інформаційних технологій “Сучасні інфокомунікаційні технології”- Київ, ДУТ, 5 грудня 2019 року);
- VII Науково-технічній конференції студентів та молодих вчених факультету Інформаційних технологій “Сучасні інфокомунікаційні технології”. – Київ, ДУТ, 5 грудня 2018 року);

- VI Науково-технічній конференції студентів та молодих вчених факультету Інформаційних технологій “Сучасні інфокомунікаційні технології”- Київ, ДУТ, 25 травня 2018 року).

Структура та обсяг дисертаційної роботи. Дисертація складається зі вступу, трьох розділів, висновків, списку використаних джерел із 108 найменувань на 13 сторінках. Загальний обсяг роботи становить 189 сторінок серед яких 142 сторінок основного тексту, 47 рисунки, 12 таблиці.

РОЗДІЛ 1
АНАЛІЗ МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСІВ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ

1.1. Мікрокомп'ютерні комплекси персональної ідентифікації, зберігання даних і особливості кодування інформації

Одним із прикладів інтенсивного розвитку інформаційного суспільства на початку XXI століття є повсюдне впровадження та поширення портативних систем, максимально наближених до власників - чіп-карт.

На сьогоднішній день чіп-карта, або карта з інтегральної мікросхемою, це пластина з полімерного матеріалу, певних розмірів. Особливість її полягає в можливості розмістити в карті звичайної товщини одну або кілька інтегральних мікросхем і в застосуванні перехідною (сполучної) плати, здатної забезпечити електричний контакт зі спеціальним перехідним хост-комп'ютером [1].

Даний клас портативних систем в даний час реалізує вкрай важливу функцію, він є носіями виняткових даних власника. Яскравими представниками цього класу є, наприклад, банківська карта, електронний паспорт, SIM-карта мобільного апарату, смарт карти. З найбільш останніх реалізацій - біометричний паспорт. Очевидно, що цінність інформації в таких системах набагато перевищують їх ринкову вартість. Втрата або спотворення інформації є критичними і не допустимі для них.

Під *мікрокомп'ютерним комплексом персональної ідентифікації* (МКПІ) в роботі розуміється сукупність пристроїв введення-виведення, блоків оперативної пам'яті і довготривалого зберігання, мікропроцесорних пристроїв і пристроїв обробки інформації, що знаходиться під єдиним сполученням і управлінням, і виконує функції ідентифікації користувача, довготривалого зберігання і своєчасного надання персональної інформації. Конструктивно МКПІ складаються з персональних ідентифікаторів,

інтерфейсів (каналів зв'язку) і хост-комп'ютерів. Необхідно відзначити, що компоненти МКПІ виконують перераховані функції в єдиній системі, ядром якої і визначальним компонентом є персональний ідентифікатор.

Під терміном *хост-комп'ютер* в роботі розуміється комплекс технічних засобів для автоматичної обробки інформації, що міститься в персональних ідентифікаторах, і надання користувачеві прикладних функцій. У МКПІ вид хост-комп'ютера визначає персональний ідентифікатор: для SIM-карти це мобільне обладнання (ME), для банківської карти - DIP-зчитувач карт (Dual In-line Package) в банківському автоматі (ATM, Automated Teller Machine) або POS-термінал касового апарату (Point OfSale, точка продажу), для RFID-міток доступу (Radio Frequency Identification, радіочастотна ідентифікація) - пристрій читання IC-карт індукції (Integrated Circuit, інтегральна мікросхема). Слід зазначити, що без відповідного персонального ідентифікатора користуватися хост-комп'ютерами неможливо і не має практичного сенсу. Хост-комп'ютери є шлюзами між користувачами та серверами баз даних. Вони переводять дані між цими двома компонентами таким чином, щоб вони могли розуміти один одного.

Персональний ідентифікатор (ПІ) машино зчитуваний носій інформації, призначений для ідентифікації власника, підтвердження відповідних повноважень, біометричних даних та інших відомостей на хост-комп'ютері. ПІ включає в себе блоки оперативної і постійної пам'яті, пристрої введення-виведення, центральний процесор і ін. З огляду на різноманіття персональних ідентифікаторів, методів зчитування і порядку ідентифікації, в рамках дисертаційного дослідження необхідно мінімізувати, або, де це можливо, виключити зміни програмного забезпечення і структури хост-комп'ютерів, переклавши функції забезпечення підвищення надійності зберігання інформації на операційну систему персональних ідентифікаторів шляхом модифікації обробки та зберігання даних. Однак, існують персональні ідентифікатори, що не містять інтегральних мікросхем. Наприклад, в картах з магнітною смугою (далі магнітні картки) носієм

інформації є магнітна стрічка. У дисертаційному дослідженні розглядається модифікація алгоритмів і методів обробки, зберігання та введення-виведення інформації для даних ідентифікаторів. Ця модифікація буде здійснюватися спільно з хост-комп'ютером [3].

Клас МКПІ об'єднує багато систем, що відрізняються за особливостями функціонування, структуру зберігання файлів, наявності або відсутності операційної системи, складності елементної бази, обсягами оперативної і постійної пам'яті. Дані обставини диктують неможливість впровадження універсальних алгоритмів і методів обробки, зберігання і введення виведення даних для всіх представників класу МКПІ. Таким чином, наприклад, магнітні карти фізично обмежені в застосуванні складних алгоритмів завадостійкого кодування, в свою чергу, персональні ідентифікатори, що містять інтегральні мікросхеми, створені за технологією SOC (System On Chip, система на кристалі), мають потенціал застосування алгебраїчних складних алгоритмів обробки, зберігання і введення-виведення інформації, оскільки підтримуються потужної операційною системою (COS) і мають достатні обсяги пам'яті. SMART-карти анонсовані в 2010-2012 рр. мають до ста мегабайт пам'яті [3].

Оскільки ядром МКПІ є саме персональний ідентифікатор, то доцільним видається вирішувати задачу класифікації МКПІ на підставі апаратно-програмної платформи і особливостям зберігання персональних ідентифікаторів.

SMART-карти, особливості кодування інформації

SMART-карти мають зону зіткнення, що містить кілька невеликих контактних пелюсток. Коли карта вставляється в хост-комп'ютер, чіп стикається з електричними конекторами, і можливо вважати або записати інформацію на чіп. SMART-карти з мікропроцесором не містять батарейок; енергія підтримується зчитувачами. Найбільш масові SMART-карти з мікропроцесором: SIM-карти, банківські картки, транспортні карти, карти доступу до цифрового телебачення, електронні ключі доступу (e-token).

Форма карти, контактів, їх розташування і призначення регламентуються в стандартах ISO / IEC 7816 та ISO / IEC 7810. Стандарт ISO / IEC 7816 регламентує також протоколи обміну і деякі аспекти роботи з даними, які використовуються і для інших SMART-карт.

Мікро-чіп карти залитий в пластмасовому корпусі типу STANDART, який має для всіх карт однакові розміри [10]. Від усіх інших карт відрізняються тільки SIM-карти типу PLUG IN. Ці карти також мають розміри типу STANDART, проте в них навколо мікро-чіпа є перфорація спеціальної форми, що дозволяє при необхідності витягти мікро-чіп з карти і вкласти в апарат стільникового зв'язку - хост-комп'ютер, який спеціально сконструйований для SIM-карт такого типу. Зовнішній вигляд і розміри пластикової карти STANDART наведені на рис. 1.1.

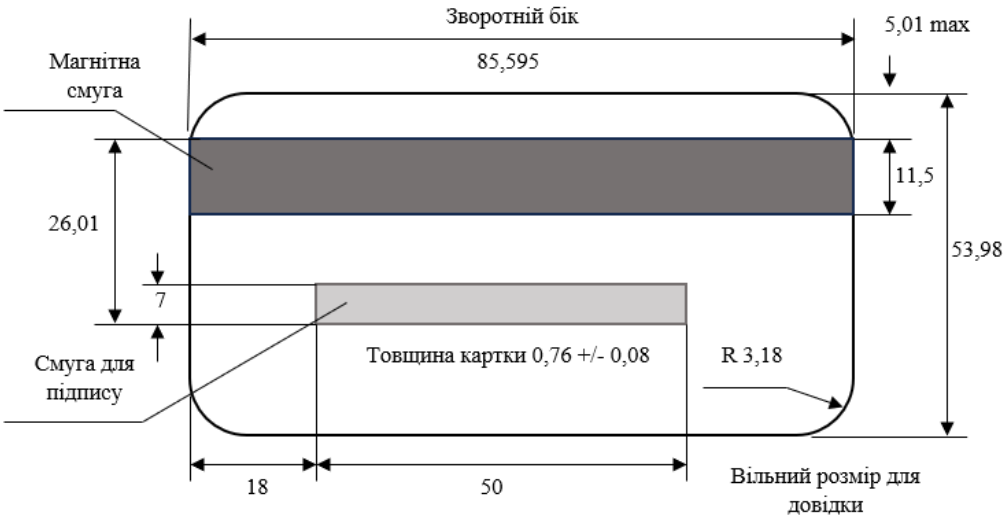

Рис 1.1. Зовнішній вигляд і розміри пластикової карти типу STAND ART

Зовнішній вигляд і розташування контактів на SIM-карті типу PLUG-IN наведені на рис. 1.2.

На поверхні SIM-карти розташовані кілька контактних майданчиків [10]. Рисунок цих контактів може бути різним, проте їх взаємне розташування і кількість повинні відповідати вимогам міжнародного

стандарту ISO 7816. Відповідно до умов, встановлених цим стандартом, кількість таких контактів має становити вісім. При цьому в сучасних SIM-картах поки використовуються лише шість контактних майданчиків.

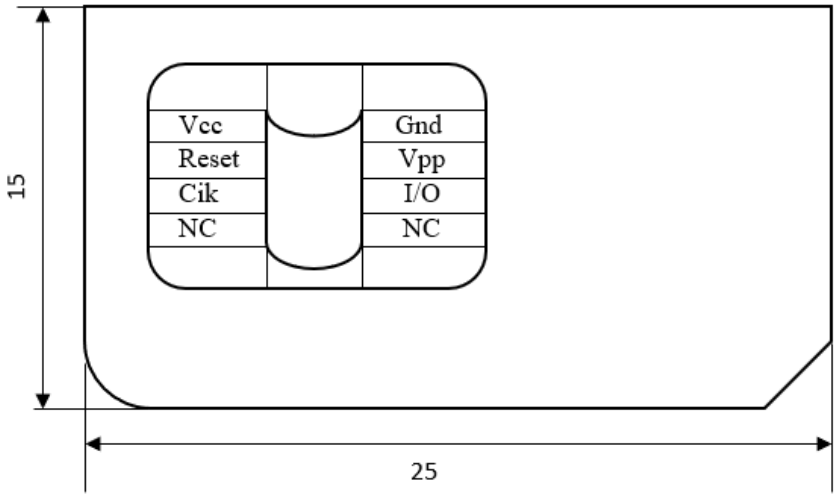

Рис. 1.2. Зовнішній вигляд і контактна площадка SIM-карти типу PLUG-IN

Стандарт ISO 7816 під назвою: «Карти ідентифікаційні. Карти на інтегральних схемах з контактами »складається з чотирнадцяти частин [1]. Для цілей дисертаційної роботи необхідно розглянути основні організаційні структури файлової системи і команди доступу та обміну інформацією зі SMART-картою. Зупинимося детальніше на стандарті ISO 7816-4, оскільки саме в ньому описані ці особливості.

Правильний аналіз інформації з ATR налаштовує хост-комп'ютер на роботу з картою. SMART-карта з інтегрованим чіпом містить процесор і найбільш важливі компоненти мікроконтролера - різні типи пам'яті, які служать для зберігання програмного коду і даних. Оскільки архітектура SMART-карти частково повторює структуру персонального комп'ютера, вона володіють характерним поділом пам'яті на оперативні запам'ятовуючі пристрої (ОЗП, RAM), постійні запам'ятовуючі пристрої (ПЗП, ROM) і електрично-перепрограмовані ПЗП (ЕППЗП, EEPROM). Точний поділ дуже сильно залежить від кінцевої області застосування чіпа. Виробники

намагаються якомога менше розміщувати RAM- і EEPROM-модулів на чіпі, оскільки вони вимагають більше площі на біт [3]. Виходячи з функціональних особливостей різних типів пам'яті в класифікації, можна зробити висновок про доцільність обмежити застосування методів і алгоритмів обробки і зберігання інформації енергонезалежною пам'яттю (EEPROM, Flash EEPROM, FRAM). Розглянути можливість установки мінімальної одиниці блочного кодування одну сторінку пам'яті об'ємом 64-128 байт.

Розглянемо структуру та можливості файлової системи SMART-карт. Як макет дослідження зупинимося на SIM-картах, які в даний час є найбільш масовими SMART-картами з мікро контролером і включають в себе продуктивний чіп і пам'ять великого обсягу. Оскільки SIM-карта - асинхронна чіп-карта, відповідна одночасно специфікаціям не тільки стандарту ISO 7816, а й стандарту GSM 11.11 [14], то для повноти відомостей, інформація буде скомпільована з декількох джерел. Стандарти багато в чому перетинаються, тому будуть описані лише ключові особливості, необхідні для визначення можливостей файлової системи, командного процесора, а так само особливостей зберігання даних.

SIM-карта являє собою мікрокомп'ютер на базі 8-розрядного процесора [15]. Мікропроцесор SIM-карти управляється більш ніж 20 командами, серед яких є команди запису / читання / пошуку файлів, виконання ідентифікації, підтримки шифрування і запуску криптографічних алгоритмів та інші. Файлова система SIM-карт має назву GSMFS. Складається з кореневої директорії і файлів, об'єднаних в ієрархічному порядку. В SMART-картах можуть бути використані тільки стандартизовані файлові структури, які визначаються стандартом ISO 7816.

Файлова система SIM-карти підтримує дві категорії файлів [9]:

- призначений файл (DF) - файл, що містить контрольну інформацію файлу і вільну пам'ять для розподілу, є «картотекою» файлів (аналог папки);

- елементарний файл (EF) - набір записів, які спільно використовують один ідентифікатор файлу. EF містять в собі дані, які використовуються операційною системою і прикладними програмами SMART-карти.

DF в підставі ієрархії називається головним файлом (MF). MF є обов'язковим. Інші DF є необов'язковими. MF є кореневим файлом і повинен бути присутнім в одиничному числі в будь-якій SMART-карті. Логічну організацію даних в карті можна уявити згідно рис. 1.3.

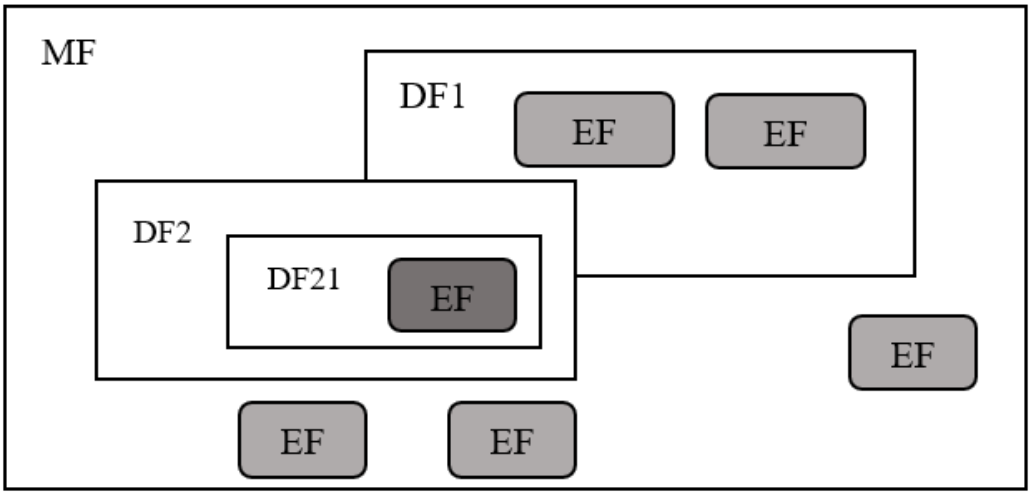

Рис. 1.3. Логічна організація файлової системи SIM-карт

Звернення до будь-якого файлу може бути здійснено за допомогою ідентифікатора файлу, кодованого в двох байтах. Якщо, наприклад, через ідентифікатор файлу здійснюється звернення до MF. то повинно використовуватися значення 0x3F00 (зарезервоване значення). Шлях починається з ідентифікатора MF або поточного DF і закінчується ідентифікатором обраного файлу. Між цими двома ідентифікаторами шлях складається з ідентифікаторів послідовних (в рамках ієрархії) батьківських EP, якщо вони є. Порядок проходження ідентифікаторів файлів - завжди в напрямку від батьківського файлу до дочірнього. Якщо ідентифікатор поточного DF не відомий, на початку шляху може використовуватися значення 0x3FFF (зарезервоване значення). Використання шляху дозволяє здійснювати однозначний вибір будь-якого файлу з MF або з поточного DF.

Кожен файл ідентифікується двома байтами. Перший байт визначає тип файлу:

0x3F: Головний файл;

0x7F: Призначений файл;

0x2F: Елементарний файл в MF;

0x6F: Елементарний файл в DF.

В даний час стандартами визначені два DF [9], [14]:

- DF_{GSM} містить додатки для GSM;
- $DF_{TELECOM}$ містить телекомунікаційні сервіси.

EF складається з заголовка і тіла. У GSM використовуються три структури побудови EF.

Прозорий EF є єдиним неструктурованим блоком пам'яті, дані в якому організовуються зручним чином для додатка, що працює з ним (рис. 1.4). Складається з послідовності байтів. У першого байта прозорого EF адреса 0x0000. Повна довжина даних тіла EF позначена в заголовку EF.

Рис. 1.4. Прозорий елементарний файл

Лінійний EF із записами фіксованої довжини ділиться на частини однакової довжини, звані записами. Записи нумеруються послідовно і доступ до них організовується за номерами або послідовно від молодшого номера до старшого або від старшого до молодшого (рис. 1.5). У заголовку EF міститься значення, яке дорівнює кількості записів помножене на їх довжину.

Рис. 1.5. Лінійний елементарний файл

Існує так само лінійний EP із записами змінної довжини, ділиться на пронумеровані частини довільної довжини.

Циклічний EF із записами фіксованої довжини є лінійним файлом, циклічно повертається до першого запису, коли проводиться запис або читання запису, наступного за останній (рис. 1.6). Циклічний формат використовуються для того, щоб зберегти записи в хронологічному порядку. Якщо розмір файлу стає максимальним, то черговий запис перезаписує найстарший. EF з циклічною структурою складається з постійного числа записів фіксованої довжини. Останній запис містить найновіші дані, є номером запису 1.

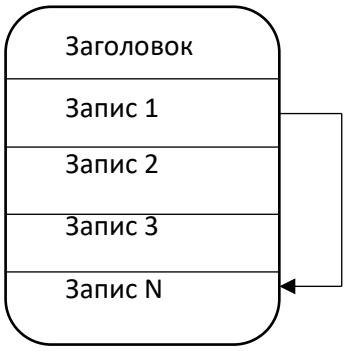

Рис. 1.6. Циклічний елементарний файл

У структурах, заснованих на записах, є наступні обмеження: записів не може бути більше 255 штук, довжина запису не може становити більше 255 байт.

Структура FSGSM і ідентифікатори всіх файлів відповідно до стандарту ETSI GSM 11.11 і ISO 7816 показані на рис. 1.7.

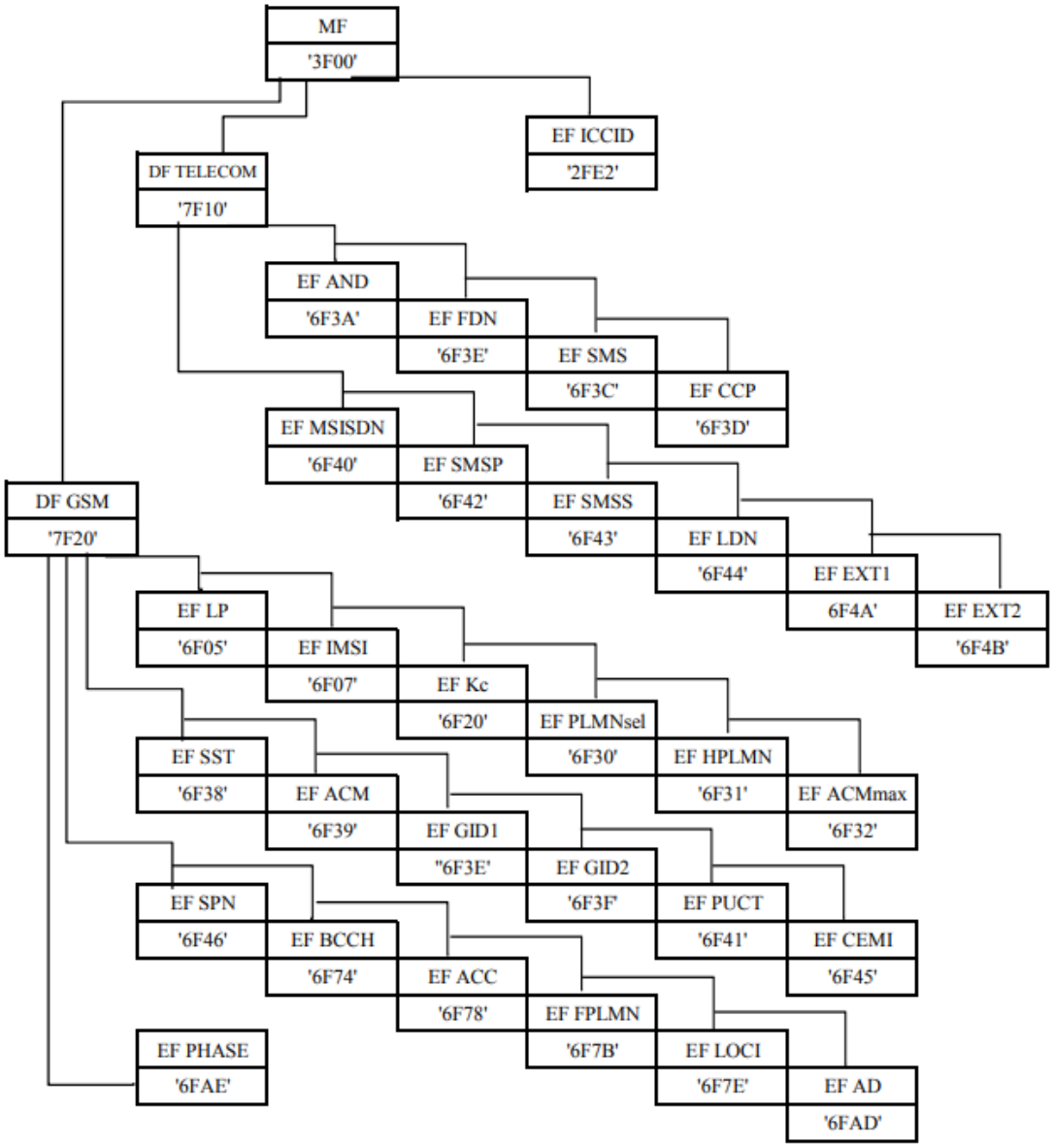

Рис. 1.7. Структура файлової системи SIM-карт

Після процедури ATR поточною Директорією призначається MF. Подальший вибір файлів може здійснюватися командою SELECT. З метою деталізації предметної області дослідження розглянемо порядок кодування типових інформаційних структур в діючих версіях операційних систем SIM-карт без конструкцій, що підвищують стійкість до помилок. Серед усіх можливих елементарних файлів зупинимося на наступних:

- EF SMS (Elementary File Shortmessages, елементарний файл, який містить короткі повідомлення)

- EF IMSI (Elementary FileInternational Mobile Subscriber Identity, елементарний файл, який містить IMSI)

- EF ADN (Elementary File Abbreviated Dialling Number, елементарний файл, який містить короткі номери)

- EF ICCID (Elementary File Integrated Circuit Card ID, елементарний файл, який містить ідентифікатор картки з інтегральною схемою)

- EF SMSP (Elementary File Last Short messages Parameters, елементарний файл, який містить параметри служби коротких повідомлень)

Елементарні файли EF IMSI і EF ICCID мають схожу структуру і кодування. Обидва файли є прозорими. Вміст EF ICCID в аналізованої SIM-карті (10 байт) представлено в 16-річній системі числення `98079109130106146 2F4`. Щоб отримати реальний номер ICCID необхідно поміняти місцями суміжні октети і прибрати останній символ-заповнювач «F» [4]. В результаті вийде: `8970199031106041264`. Вміст EF IMSI (9 байт): `082905995710561262`, представлене в 16-річній системі числення. Щоб отримати реальний номер IMSI необхідно поміняти місцями суміжні октети. В результаті вийде: `082905995710561262`.

Елементарний файл, який містить параметри служби коротких повідомлень, має лінійну структуру і складається з послідовності записів, наведених у таблиці 1.1.

Таблиця 1.1.

Кодування запису елементарного файлу SMS-центру

Байт	Опис	М/О	Довжина (байт)
1-X	Альфа ідентифікатор	O	X
X+1	Індикатор параметрів	M	1
X+2- X+13	Адрес призначення	M	12

X+14-X+25	Адрес сервіс центра	M	12
X+26	Ідентифікатор протоколу	M	1
X+2 7	Схема кодування	M	1
X+28	Період оновлення	M	1

Інформація з елементарного файлу, що містить параметри служби коротких повідомлень, виглядають наступним чином: `34D53432031E1FFFFFFFFFFFFFFFFFFFFFFFFF07919730071111F1DC1F31A80000A7`

Очевидно, номер SMS-центру знаходиться в діапазоні між 28-им і 35-им байтами: `07919730071111F1`. Перший байт містить довжину телефонного номера. Другий байт описує тип номера. Щоб отримати реальний номер SMS-центру необхідно поміняти місцями суміжні октети (починаючи з третього байта) і прибрати останній символ-заповнювач «F» [4].

Елементарний файл SMS містить інформацію про короткі повідомлення і пов'язані параметри, які були отримані з мережі. Файл має циклічну структуру і складається із записів в хронологічному порядку. Кожен запис описує окреме SMS повідомлення. Структура запису наведена в таблиці 1.2.

Таблиця 1.2.

Кодування запису елементарного файлу SMS-повідомлень

Байт	Опис	Довжина (байт)
1	Статус	1
2-9	Номер SMS центра	8
10	Модифікатор повідомлення	1
11-18	Номер абонента	8
20-21	Тип кодування тексту	2
22-29	Дата прийому	7

30	Довжина повідомлення	1
31-176	Текст повідомлення	145

Розглянемо порядок кодування інформації в файлі на прикладі. Перші 40 байт першого запису EF SMS, в 16-річній системі числення представлені з описом в таблиці 1.3.

Таблиця 1.3.

Кодування запису в файлі SMS-повідомлень

01	Статус повідомлень
07 9197 6198 9901F0	Номер SMS центра
24	Модифікатор повідомлення
0B919761349066F80	Номер відправника
0000	Тип кодування тексту
60808080736161	Час и дата прийому\відправки
9E	Кількість байт в повідомленні
C4B758FE2EB3F4...	Текст повідомлення (7 байт)

Першим байтом запису є байт статусу, він описує тип повідомлення входить / виходить, прочитане / непрочитане, актуальне / віддалене. Далі, слідує номер SMS центру, через яке повідомлення відправлялося або приймалося. Модифікатор повідомлення дорівнює 0x24. Це означає, що повідомлення складається з одного. Якщо дорівнює 0x60, то повідомлення складається з декількох, при цьому в тексті кожного такого повідомлення міститься унікальний ідентифікатор набору повідомлень, а повідомлення в межах набору нумеруються послідовно.

Номер відправника розпізнається аналогічно EF IMSI. і, в даному прикладі, дорівнює `79164309668`. Тип кодування вказує на схему кодування тексту повідомлення. Якщо 0x0000, то це GSM кодування. Кожен символ при цьому кодується 7-ю бітами. Якщо 0x0008, то кодування UNICODE, при цьому кожен символ кодується 16-ю бітами. Наступні 7 байт

описують дату і час відповідно до стандарту [16]. Щоб отримати реальні дату і час, необхідно поміняти місцями суміжні октети. Результат розпізнавання в таблиці 1.4.

Таблиця 1.4.

Кодування дати та часу

Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
Рік	Місяць	День	Година	Хвилина	Секунда	Часовий пояс
0x19	0x08	0x08	0x08	0x37	0x16	0x16
2019	Серпень	08	08	37	16	+4

Таким чином, повідомлення датоване 8 серпня 2019 року в 8:37:16 GMT +2. Текст повідомлення розпізнається відповідно до стандарту [17]. Кожні 7 байт: `C4 B7 58 FE 2 E B3 F4` кодують 8 символів латинського алфавіту. Кожен символ кодується 7 бітами. При цьому старший біт першого байта є молодшим бітом другого символу, а молодші 7 біт першого байта утворюють перший символ. Далі старші два біти другого байта є молодшими бітами третього символу і так далі. У таблиці 1.5. показаний приклад розпізнання перших 7 байт тексту повідомлення. У стовпці «Значення (2)» позначені біти, що доповнюють значення наступних символів. Зауважимо, що сім старших біт сьомого байта повністю складають восьмий символ «у».

Таблиця 1.5.

Кодування тексту SMS-повідомлень

Номер байта	Значення(16)	Значення (2)	Залишок	Символ
1	0xC4	**1**100 0100		D
2	0xB7	**10**11 0111	1	O
3	0x58	**010**11000	10	B
4	0xFE	**1111** 1110	010	Г
5	0x2E	**0010 1**110	1111	O
6	0xB3	**1011 00**11	00101	E
7	0xF4	**1111 010**0	101100	,

			1111010	Y

Особливості кодування тексту SMS-повідомлення необхідно враховувати при формуванні розміру блоку для завадостійкого коду.

Циклічний елементарний файл EF ADN містить останні набрані номери, ідентифікатори мережі і ідентифікатори додаткових записів (порядок кодування запису наведено в табл. 1.6).

Таблиця 1.6.

Запис елементарного файлу останніх набраних номерів

Байт	Опис	M/O	Довжина(байт)
1 -X	Альфа ідентифікатор	O	X
X+1	Довжина 88C	M	1
X+2	Тип номера	M	1
X+3 -X+12	Набраний номер	M	10
X+13	Ідентифікатор конфігурації	M	1
X+14	Допоміжний ідентифікатор запису	M	1

Приклад першого запису EF ADN (39 байт), представлений в 16-річній системі числення.

`81070890BBD1BDB02F4DFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0 79197305175 55F2FFFFFFFFFFFF`

Запис складається з імені контакту `81070890BBD1BDB02F4D` і заповнювача, та номеру телефону `07919730517555F2` і заповнювача.

Телефонний номер розпізнається аналогічно EF IMSI і в даному прикладі дорівнює 79031557552. Ім'я абонента записано за допомогою стандартного алфавіту GSM [17]. Якщо першим символом запису є байт 0x81, то другий байт буде містити значення, яке вказує на кількість знаків у рядку. Третій байт містить номер, що задає біти з 15 по 8 16-бітного базового покажчика на сторінку UNICODE, де біт 16 і біти з 7 по 1 матимуть значення

0. Четвертий і наступні байти рядки кодуються так: якщо біт 8 має значення 0, ті, що залишилися біти містять символи стандартного алфавіту GSM. Якщо біт 8 має значення 1, то ті, що залишилися біти будуть зміщенням щодо 16-бітного базового покажчика, заданого байтом 3. Результуюче 16-бітове значення має кодування UNICODE і визначає символ UNICODE.

У багатьох файлах з метою підтримки стандартної довжини, використовуються символи-заповнювачі 0xFF. Очевидно висновок, що на місці байт-наповнювачів могла бути корисна інформація для відновлення запису в разі виникнення помилок.

Для обміну інформацією з SMART-картою використовуються блоки даних прикладного протоколу (APDU, application protocol data unit), де зазначено команда, або відповідь була надіслана, що циркулює між картою і хост-комп'ютером [12], [18]. Опис методів обміну інформацією з хост-комп'ютерами для збереження наступності з новою версією SMART-карти в процесі модернізації файлової системи і заміни стандартних методів читання і запису на перешкодостійкі з метою уникнути економічно недоцільною модернізації хост-комп'ютерів.

Таким чином, SMART-карти з мікропроцесором це найбільш технічно-складні представники класу МКПІ. Є повноцінними обчислювальними комплексами, що задовольняють вимогам стандарту ISO 7816. Дані SMART-карти мають у своєму розпорядженні гідними обсягами пам'яті, операційною системою, ієрархічно організованою файлової системою. При дотриманні протоколу взаємодії з хост-комп'ютером, збереженні структури блоків АРПП і процедур обміну блоками на прикладному рівні, в даних пристроях можливо найбільш ефективне й економічно не витратне впровадження алгоритмів підвищення надійності, оскільки зміна торкнеться виключно файлову систему і не спричинить зміни різних хост-комп'ютерів.

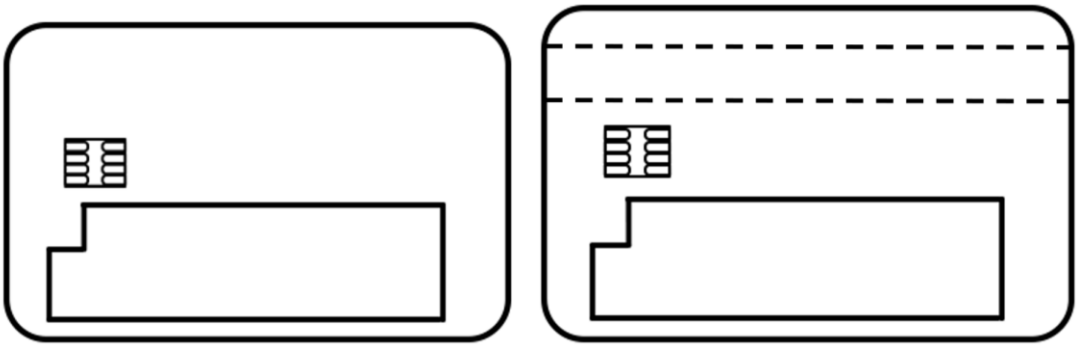

Рис. 1.8. Розташування магнітної смуги на SMART-картах з мікропроцесором

Відзначимо, що деякі SMART-карти з мікропроцесором містять так само область з інформацією, записаною на магнітну смугу, частково відтворює вміст блоків пам'яті мікропроцесора. Магнітна смуга, як правило, містить таку нормативну інформацію, доступну для читання відповідним хост-комп'ютером. Прикладом SMART-карти з магнітною смугою і мікропроцесором одночасно є деякі банківські карти.

Зберігання та кодування даних у магнітних картах.

Магнітні картки, як носій інформації, використовують магнітну смугу. Магнітна смуга може бути виготовлена з матеріалів з різною силою магнітного поля. За цим параметром смуги поділяються на низько- і високо-коерцитивні. Ступінь коерцитивності впливає стійкість записаної інформації до розмагнічування. Чим вища коерцитивність, тим складніше розмагнітити смугу магнітного кодування. Топологія Міжнародної організації зі стандартизації (ISO) виділяє для магнітної лінії місце вгорі карти на звороті. У випадках, коли на магнітну смугу записано до 100 байт, ця інформація дублюється штрих-кодом внизу карти. Принципова різниця між SMART-картою з мікропроцесором і карткою з магнітною смугою - це рівень безпеки, який карта дає своєму власнику. Читати інформацію з магнітної лінії і розшифрувати її легко. Крім того, SMART-карти з мікропроцесором не так схильні до пошкоджень, як магнітна смуга. Незважаючи на ці недоліки, сфера застосування карток з магнітною смугою дуже велика. Стимулює таке

поширення їхня невисока вартість та простота виготовлення. Це можуть бути банківські картки, картки лояльності, транспортні картки, картки доступу.

Магнітна смуга складається з трьох треків - смуг, що відповідають міжнародному стандарту ISO 7813, 7811 та званих ISO1, ISO2 та ISO3. Раніше в картах типу VISA були і додаткові смуги T2 і T3 [1], рис. 1.9.

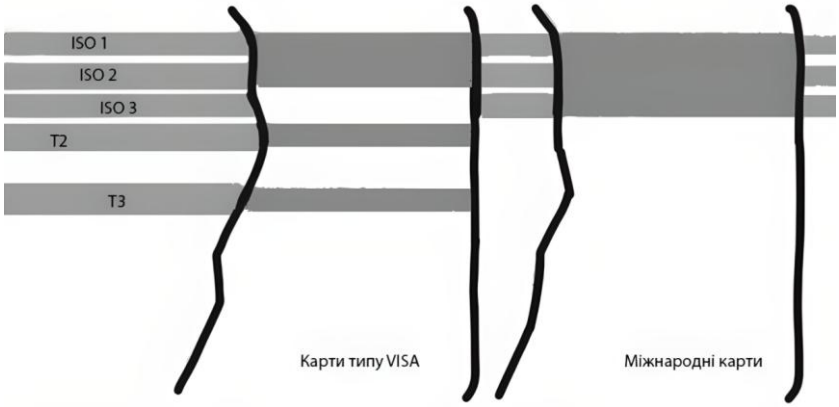

Рис. 1.9. Розташування магнітних смужок на картці

Згідно схеми намагнічування за стандартом ISO 7813 перша смуга може містити до 76 символів, закодованих 7 бітами кожен (код ASCII) і щільністю запису 210 біт/дюйм. Асоціація AAMVA дозволяє записати 82 семи бітові символи з тією ж щільністю на дюйм. Друга смуга має щільність запису 75 біт/дюйм, що рівносильна 40 п'яти бітовим числовим символам. Смуга IS ОЗ має щільність аналогічну ISO1 210 біт на дюйм і містить 107 символів.

S	C	AN	S	AME	S	ADDITIONAL DATA	DISCRETIONAL DATA	ES 1LRC

Рис. 1.10. Смуга ISO1

Смуга ISO1 використовується для електронних квитків та транспортних карток. Структура лінії показана на рис. 1.10 і включає:

- SS: Прапор початку, символ "%"
- FC: Формат, символ A-Z
- PAN: Головний номер
- FS: роздільник полів, символ

- NAME: Ідентифікатор
- ADDITIONAL DATA: Додаткова інформація
- DISCRETIONAL DATA: Дискреційна інформація
- ES: Прапор закінчення
- LRC: Контрольна сума

S	AN	S	ADDITIONAL DATA	DISCRETIONAL DATA	S	1L RC

Рис. 1.11. Смуга ISO2

Смуга ISO2, як правило, використовується хост-комп'ютерами POS та ATM у фінансовій галузі. Структура лінії показана на рис. 1.11 і включає:

- SS: Прапор початку, OxB
- PAN: Головний номер (максимум 19 символів)
- FS: Розділювач полів, як правило, символ «=»
- ADDITIONAL DATA: Додаткова інформація
- DISCRETIONAL DATA: Дискреційна інформація
- ES: Прапор закінчення, символ "?"
- LRC: Контрольна сума (операція XOR усіх символів)

S	C	AN	S	ADDITIONAL DATA	DISCRETIONAL DATA	S	RC

Рис. 1.12. Смуга ISO3

Смуга ISO3 використовується в картах лояльності, доступна не тільки для читання, але й для запису інформації. Структура лінії показана на рис. 1.12 і включає:

- SS: Прапор початку, 0xB
- FC: Код формату (2 цифри)
- PAN: Головний номер (максимум 19 символів)
- FS: Роздільник полів, символ "="
- ADDITIONAL DATA: Додаткова інформація
- DISCRETIONAL DATA: Дискреційна інформація

- ES: Прапор закінчення, 0xF

- LRC: Контрольна сума (операція XOR усіх символів)

Кожній зміні напряму намагнічування на смузі відповідає вертикальна характеристика рис 1.13.

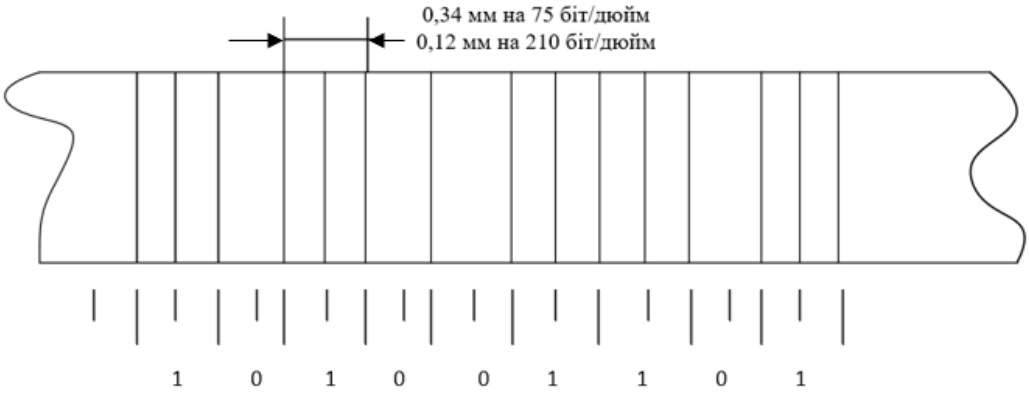

Рис. 1.13. Зберігання інформації на магнітній смузі

Таким чином, магнітні карти є найбільш схильними до різних помилок представниками класу МКПІ. Мінімальна вартість виготовлення стимулює їхнє застосування у широкому спектрі телекомунікаційних послуг. Магнітна смуга може бути більш надійною, якщо використовувати певні правила її використання, наприклад, блочне кодування, яке стійке до перешкод.

1.1.1 Мікрокомп'ютерні комплекси персональної ідентифікації на основі технології RFID

Системи радіочастотної ідентифікації тісно пов'язані з описаними чіп-картами. Тут також носієм даних є електронний пристрій, в даному випадку - транспондер. Однак подача живлення та обмін даними виконуються без будь-якого безпосереднього контакту за допомогою електромагнітного поля. Основою даної технології є методи, що набули широкого поширення в радарних та радіосистемах [19].

Комплекси Radio Frequency Identification (RFID) здатні зберігати дані та передавати їх хост-комп'ютеру безконтактним способом за допомогою радіохвиль, склад мітки показаний на рис. 1.14.

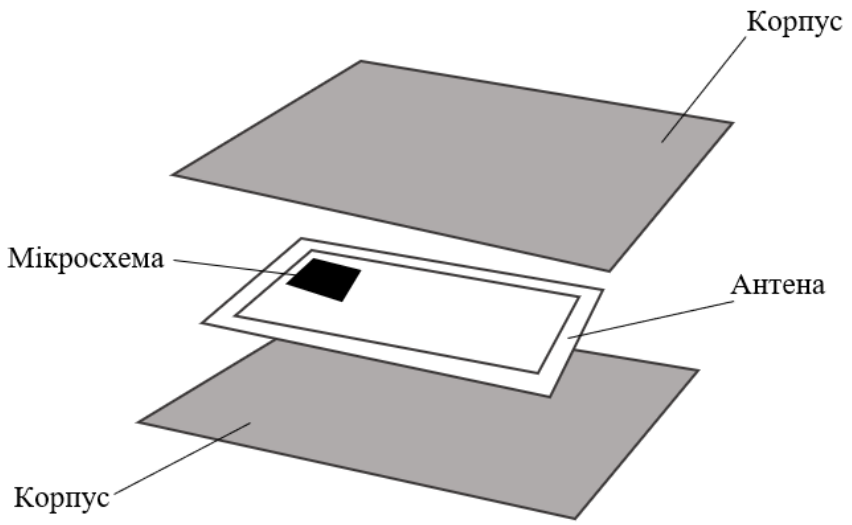

Рис. 1.14 Типовий склад RFID-мітки

Блок управління живленням перетворює напругу живлення змінного струму, що отримується від сигналу з антени хост-комп'ютера (рідера), в напругу живлення постійного струму [21]. Цей блок подає живлення в інші компоненти мікрочіпа. Детектор тактової частоти витягує тактовий сигнал із сигналу, що отримується від антени рідера. Модулятор модулює сигнал, що отримується від рідера. Модульований сигнал вводить відповідь мітки, і цей сигнал потім передається назад рідеру.

Загальний принцип роботи будь-якої RFID-системи показано на рис. 1.15 [20]. У системі завжди є три основні компоненти: це зчитувач (рідер), персональний ідентифікатор (карта, мітка, брелок, тег) та комп'ютер. Зчитувач випромінює в навколишній простір електромагнітну енергію. Ідентифікатор приймає сигнал від зчитувача і формує сигнал у відповідь, який приймається антеною зчитувача, обробляється його електронним блоком і по інтерфейсу направляється в комп'ютер або інший акумулюючий пристрій.

Завданням RFID-системи є зберігання інформації про об'єкт із можливістю її зручного зчитування. Мітка може містити дані про тип об'єкта, вартість, вагу, температуру, дані логістики, взагалі будь-яку інформацію, яка може зберігатися в цифровій формі.

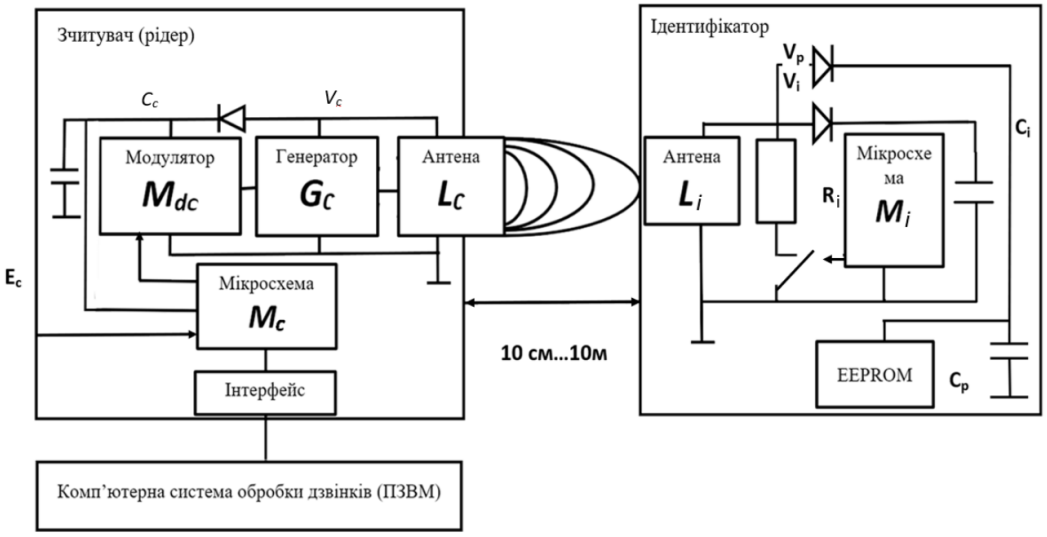

Рис. 1.15. Принцип роботи RFID-системи

Останнім часом завдяки своїм очевидним перевагам над іншими системами автоматичної ідентифікації системи RFID завойовують дедалі більшу частку ринку. Наприклад, вони все частіше застосовуються у вигляді безконтактних чіп-карток для оплати проїзду у громадському транспорті [19]. Технології радіочастотної ідентифікації застосовуються в проксіміті-картах, чіп-ключах автомобіля, паркувальних автоматах, промисловості, транспортної та складської логістики, медицині.

У контексті підвищення надійності МКПІ радіочастотної ідентифікації розглянемо різновиди зберігання інформації у пам'яті радіочастотної мітки. Спосіб запису інформації залежить від конструктивних особливостей мітки. Залежно від цього розрізняють такі види:

- RO – мітки, які працюють лише на зчитування інформації (readonly). Мітка RO-типу може бути запрограмована лише один раз у своєму життєвому циклі. Дані заносяться в мітку постійно в заводських умовах тільки на етапі її виготовлення. Для цього в мікрочіпі мітки за допомогою тонкого лазерного променя перепалюються окремі перемички [9]. Після того, як це зроблено, дані не можна буде перезаписати протягом усього терміну служби мітки. RO-мітки знаходять застосування там, де не потрібно

передавати великі обсяги даних або просто потрібно замінити традиційні системи штрихового кодування.

- WORM - мітки для одноразового запису та багаторазового зчитування інформації (write once read many). WORM-мітка може бути запрограмована одноразово, і це робиться користувачем мітки в той час, коли необхідно створити мітку. На практиці для виправлення помилок при такому записі допускається можливість перезапису деяких типів WORM-міток [5].

- RW - мітки багаторазового запису та багаторазового зчитування інформації (readand write). RW-мітка може бути перезаписана 100 000 разів і більше [7]. У мітці RW-типу зазвичай міститься пристрій пам'яті Flash- або FRAM-типу. Очевидно, мітки такого типу найдорожчі у виготовленні. Дані мітки є найбільш перспективними в даний час і в міру вдосконалення технологічності та застосовності міток знижується їхня вартість.

Для повноти класифікації необхідно згадати мітки SAW-типу (surf aceacoustic wave, поверхнева акустична хвиля). Вони докорінно відрізняються від міток на основі мікрочіпів тим, що працюють на принципі відбитої поверхневої акустичної хвилі. В даний час пристрої SAW широко застосовуються в обладнанні кінцевого користувача (ОКК). Для роботи міток SAW-типу використовуються радіохвилі малої потужності в частотному діапазоні 2,45 ГГц [7]. На відміну від міток із мікрочіпами SAW-мітці не потрібне джерело постійного струму для її живлення при передачі даних.

Пам'ять персональних ідентифікаторів, реалізованих за технологією радіочастотної ідентифікації, можна розділити на дві різні групи (рис. 1.16) [22]:

- носії даних, які використовують для зберігання даних фізичні принципи;

- електронні носії даних, що базуються на інтегральних схемах.

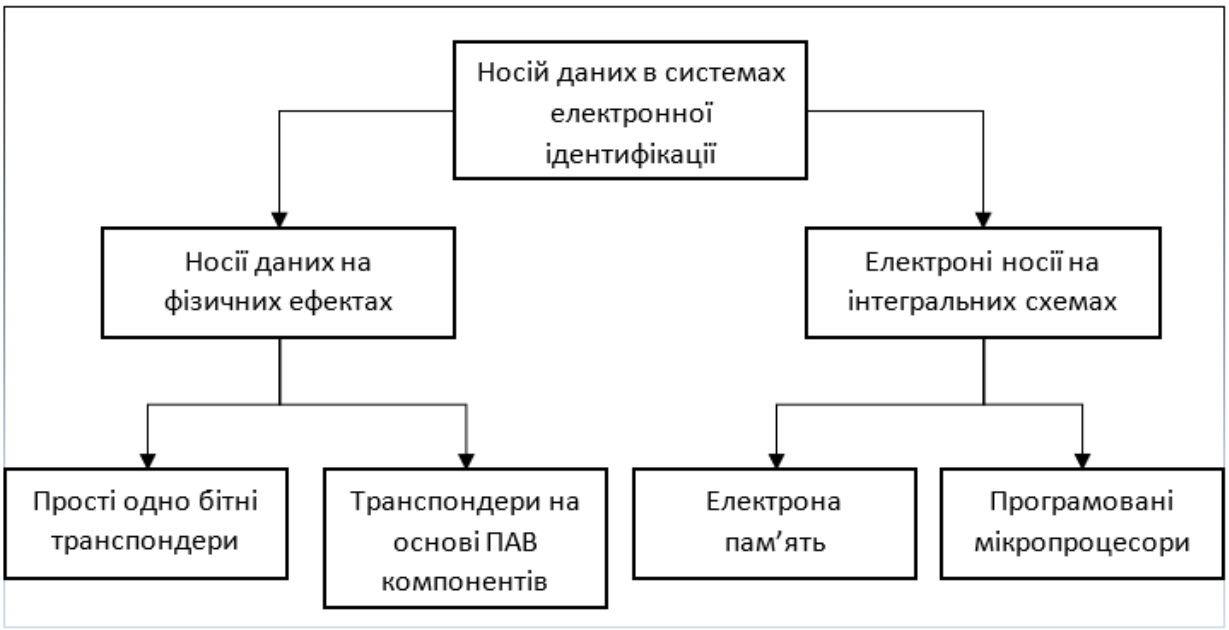

Рис. 1.16 Носії даних технології RFID

Вважається недоцільним вирішувати завдання підвищення надійності за допомогою застосування кодів корекції помилок для гілки носіїв даних на різних фізичних ефектах (рис. 1.16), однобітові транспондери та транспондери з компонентами на поверхневій акустичній хвилі не можуть повністю задовольнити вимоги до предметної області, оскільки не мають необхідних мікросхем та пам'яті, оскільки застосовуються переважно для ідентифікації матеріальних цінностей. У свою чергу, RFID-пристрої з носіями інформації, реалізованими за допомогою електронної пам'яті та програмованих мікропроцесорів мають необхідні можливості для модифікації алгоритмів обробки, зберігання та введення-виведення даних.

У системах RFID з індуктивним зв'язком, як і в SMART-картах, зазвичай використовується ЕСППЗП. Місткість пам'яті становить від 16 байт до 100 Кбайт [7]. Такі транспондери мають вбудовану схему, яка декодує досить прості команди, отримані від рідера, і керує як процесом передачі даних, так і процесом запису даних у власну пам'ять. Також застосовуються елементи фероелектричної пам'яті з довільним доступом. Витрати енергії під час виконання операції читання у FRAM нижче, ніж у ЕСППЗП з

коефіцієнтом 100, а час запису в 1000 разів нижче [22]. Доступ до пам'яті транспондера для запису чи читання даних та будь-які запити на авторизацію операцій запису та читання контролюються внутрішньою логікою носія даних. У найпростішому випадку ці функції можуть бути реалізовані кінцевим автоматом. При використанні кінцевих автоматів можна реалізувати досить складні послідовності. Недоліком кінцевих автоматів є відсутність гнучкості при внесенні змін до запрограмованих функцій. Ця проблема вирішується під час використання в електронних носіях мікропроцесорів. На стадії виробництва в мікропроцесор вводиться операційна система для управління даними програми та програмне забезпечення, адаптоване до виконання різних додатків. Використання мікропроцесорів дозволяє реалізувати досить складні алгоритми аутентифікації та шифрування переданих даних, які не можна реалізувати за допомогою фіксованої схеми на жорсткій логіці. Більшість мікропроцесорних RFID-систем використовує частоту 13.56 МГц, а протокол передачі між транспондером і хост-комп'ютером відповідає стандарту ISO 14443.

Зупинимося докладніше на RFID-мітках, що містять пам'ять мікрочіпа для зберігання даних, оскільки такі ідентифікатори є предметом дослідження в другому розділі. У цьому випадку пам'ять розбита на сегменти (тобто складається з кількох блоків чи полів). Блоки адресовані, що означає можливість звернутися до окремих ділянок пам'яті мікрочіпа. Основною характеристикою електронної начинки транспондера є обсяг пам'яті, що використовується для зберігання даних.

Розглянемо докладніше архітектуру та особливості зберігання та розподілу інформації в RFID-мітках з функцією пам'яті. Існують різні транспондери з функцією пам'яті: від простих транспондерів типу «тільки читання», до високоорганізованих транспондерів з інтелектуальними криптографічними функціями [22]. Усі вони задовольняють загальну концепцію побудови на рис. 1.17.

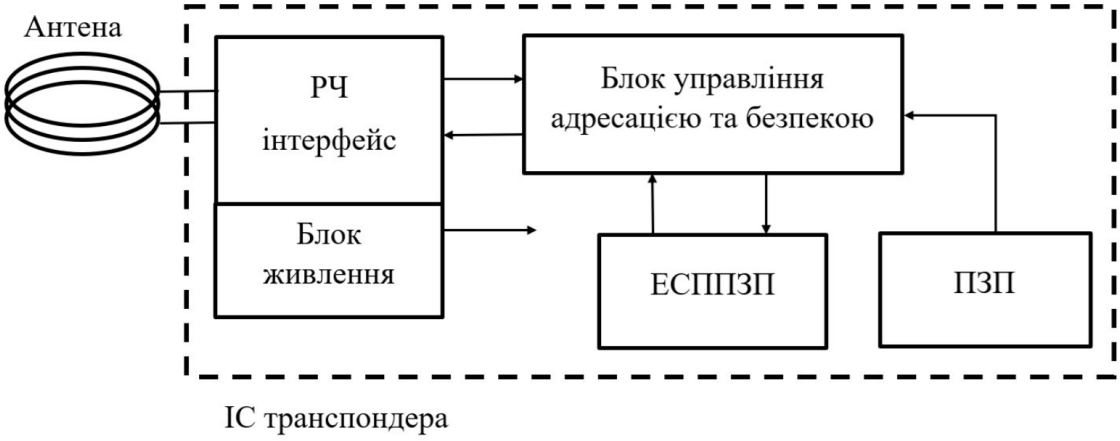

Рис. 1.17. Архітектура транспондерів із функцією пам'яті

У цій архітектурі інтерес представляє ядро транспондера - блок управління адресацією та безпекою та блоки пам'яті. Передбачається внести зміни до керуючих процесів у ядрі пристрою: модернізувати функції читання, запису, корекції помилок та вміст пам'яті при збереженні структури адресації. Пам'ять даних включає ПЗП для таких постійних даних, як порядковий номер, та ЕСППЗП або FRAM, приєднується до блоку управління адресацією та безпекою через шину адрес та даних усередині мікросхеми [23].

Розподіл пам'яті в інтегральній схемі транспондера показано на рис. 1.18. Вся пам'ять поділяється на сектори, кожен сектор може бути захищений ключами доступу та запрограмований на операції читання, запису, перезапису. Кожен сектор адресується і складається з блоків даних. Як правило, сектори містять по 4 блоки, проте існують карти із секторами по 16 блоків даних. Об'єм пам'яті одного блоку даних залежить від призначення RFID-мітки від 32 біт до 1 Кбайт. У першому блоці (на адресу 0x00) розміщуються дані ідентифікації виробника, моделі, параметри конфігурації.

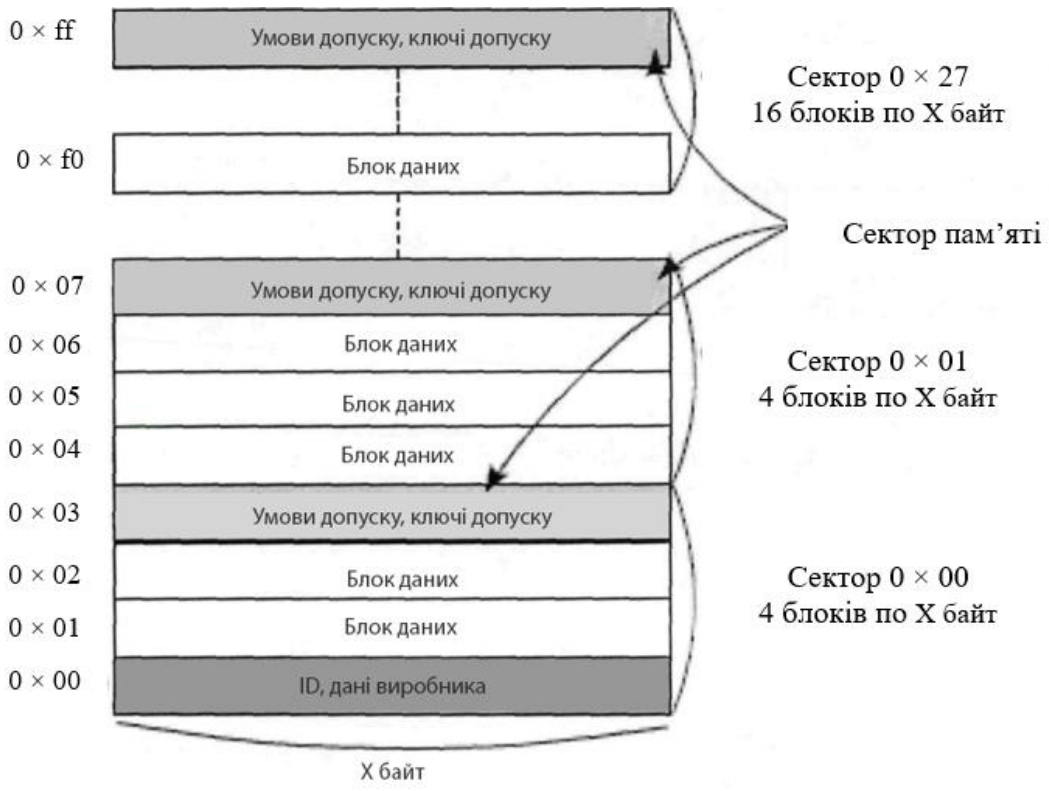

Рис. 1.18. Розподіл пам'яті в інтегральній схемі транспондера

Як приклад функціонування та реального розподілу пам'яті в інтегральних мікросхемах RFID-систем, наведемо архітектуру та карту пам'яті транспондера Atmel e5561A (рис. 1.19) [24]. Цей пристрій здатний виконувати всі функції, необхідні для іммобілайзерних та ідентифікаційних комплексів. Розташована на кристалі 80-кілобайтова пам'ять ЕСППЗП (10 блоків, по 512 біт кожен) може бути зчитана та записана побічно хост-комп'ютером. Чотири або шість блоків містять дані користувача, а два або чотири блоки пам'яті використовуються для зберігання крипто ключа, опцій читання/запису і даних ідентифікації. Крипто ключ та ідентифікаційні дані можна індивідуально захистити відперезаписом. Крім того, неможливе зчитування крипто ключа.

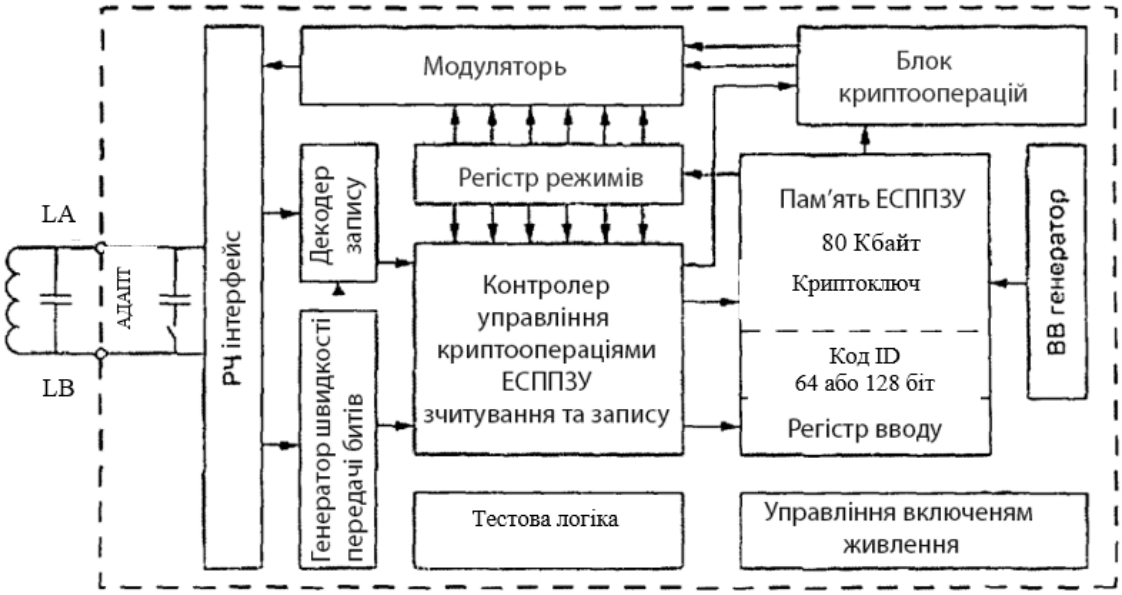

Рис. 1.19. Архітектура інтегральної схеми транспондера Atmel e5561A

Організація блоків у пам'яті ЕСППЗУ показано на рис. 1.19. Блок за адресою 0x00 використовується для заздалегідь визначених параметрів та роботи МКПІ. Наступні блоки 1-9 відведені під робочі дані швидкого доступу та вільні для програмування. Пам'ять зчитується, записується чи вибирається 512-бітовими блоками. Команда хост-комп'ютера має доступ до кожного блоку, використовуючи адресу блоку [1]. У режимі доступу паролем блок 9 містить пароль, причому перші 4 біта - дані для конфігурації користувача. Якщо пароль не активовано – відповідні біти можуть бути використані під програмування (рис. 1.20).

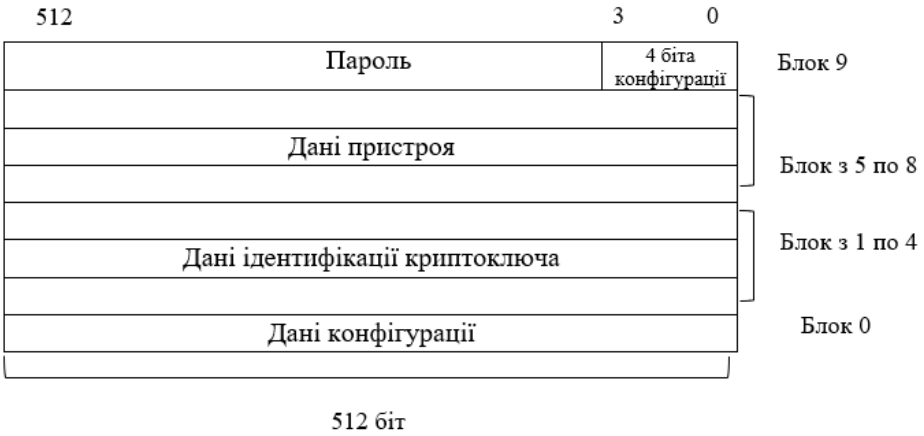

Рис. 1.20 Карта пам'яті інтегральної схеми транспондера Atmel e5561A

Таким чином, було з'ясовано, що в системах RFID з індуктивним зв'язком для зберігання найчастіше використовується від 16 байт до 100 Кбайт пам'яті EEPROM, яка забезпечує до 100 000 циклів запису даних. Раніше також використовувалася і пам'ять FRAM, яка в порівнянні з EEPROM має нижче енергоспоживання, а також менше запису. Однак широкому поширенню пам'яті FRAM заважають технологічні проблеми, що виникають під час її виробництва. У мікрохвильових системах RFID для зберігання даних також може використовуватися пам'ять SRAM, яка має надзвичайно швидкий цикл запису та читання даних. Однак ця пам'ять не є енергонезалежною, і для запобігання втраті даних необхідно використовувати безперервне джерело живлення, обсяг SRAM-пам'яті в RFID-мітках становить від 256 байт до 64 Кбайт. Серед різноманіття реалізацій RFID-пристроїв, нас цікавлять лише інтегральні мікросхеми з пам'яттю і програмовані мікропроцесори, оскільки в них можна без внесення змін до архітектури модифікувати алгоритми обробки, зберігання та введення-виведення даних. Пам'ять RFID-міток розбита на блоки, об'єднані в сектори, розмір блоку дозволяє розмістити в ньому стійкі до помилок кодові послідовності, а механізм об'єднання в секторі дозволить коригувати не тільки помилки бітів в окремо взятих блоках, але й блоки.

1.1.2 Біометричні паспорти як особливий персональний ідентифікатор

Дані персональні ідентифікатори виділені в окрему категорію не тільки через їх особливу соціальну значущість для власника і держави і винятковість даних, що зберігаються, а й через функціональні особливості. Електронні (біометричні) паспорти акумулюють у собі функціональні особливості контактних SMART-карток, у той же час здатні «спілкуватися» з хост-комп'ютером за допомогою технології RFID.

Біометричні паспорти це мікропроцесорні SMART-карти з RFID, або з подвійним інтерфейсом передачі даних. На сьогоднішній день вони є найбільш складними та продуктивними представниками МКПІ. Важливу

роль в електронному паспорті відіграє компоненти SMART-карти та біометричні дані власника картки, що зберігаються в ній, які використовуються для його ідентифікації. На поверхні паспорта, як правило, нанесено загальну інформацію про власника картки, таку як фотографія, прізвище ім'я та по батькові, стать, національність, рукописний підпис тощо. Ця інформація використовується лише з метою візуального інформування. Більш детальна та конфіденційна інформація знаходиться усередині, на інтегральних схемах.

МКПІ називається біометричним, тому що на підставі інформації, що знаходиться в ЕСППЗП, персонального ідентифікатора можна ідентифікувати людей. Ідентифікація проводиться за такими характеристик, як відбиток пальця, розмір і форма руки, голос і деякі характеристики ока [22].

У біометричних паспортах з індуктивним зв'язком поєднано переваги SMART-карт та RFID-систем. Безконтактні карти не вимагають будь-якого електричного з'єднання між карткою та хост-комп'ютером для того, щоб передати енергію та дані на коротку відстань [26]. Індуктивний зв'язок є широко використовуваним методом створення безконтактного інтерфейсу між SMART-картами і хост-комп'ютером. Індуктивний зв'язок може бути використаний для передачі енергії та даних. На рис. 1.21 показані основні конструктивні елементи безконтактної SMART-карти. До складу безконтактної SMART-карти входять індуктивна антена та чіп з інтегральною схемою. Інтегральної схеми чіп міститься у мініатюрний модуль, який підключається до кінців антени [22].

Незважаючи на те, що безконтактні мікропроцесорні SMART-карти часто асоціюються з більш складними додатками, вони також можуть використовуватися в додатках, які вимагають простоти використання, наприклад, у системах контролю доступу, і швидкості, наприклад, в електронних квиткових системах оплати проїзду в громадському транспорті [22].

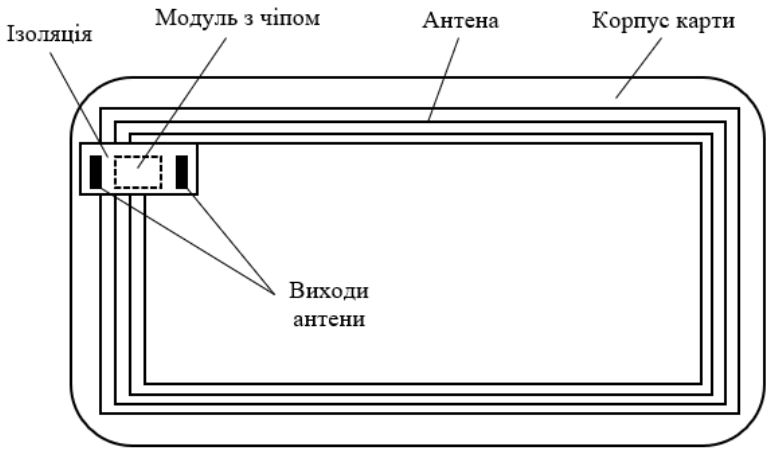

Рис. 1.21 Конструктивні елементи безконтактної SMART-картки

Розглянемо докладніше архітектуру та особливості зберігання інформації у біометричних паспортах із подвійним інтерфейсом. Тенденція, спрямована на комбінування платіжних додатків з типовими безконтактними додатками, призвела до розробки SMART-карти з подвійним інтерфейсом, в якій на одному чіпі доступні як контактний, так і безконтактний інтерфейси [22]. Відповідно, до карти з подвійним інтерфейсом можна звертатися як через контактний, так і через безконтактний інтерфейс. Принцип дії, що лежить в основі SMART-карти з подвійним інтерфейсом, полягає в тому, що інтерфейс SMART-карти є повністю незалежним від логіки та програмного забезпечення SMART-карти. Інтерфейс, контактний або безконтактний, повністю прозорий для прикладних даних, що передаються, тому з точки зору прикладного програмного забезпечення, тип використовуваного інтерфейсу є несуттєвим. Відповідно, інтерфейс замінюється за бажанням, і інтерфейс, і логічні компоненти можуть комбінуватися як завгодно [27, 28].

Типова архітектура біометричного паспорту з подвійним інтерфейсом відповідає вимогам гарвардської архітектури і показана на рис. 1.22. До кожного інтерфейсу паспорта можна звертатись незалежно від іншого. Інструкції та дані зберігаються на різних фізичних пристроях. Канал інструкцій та канал даних також фізично розділені.

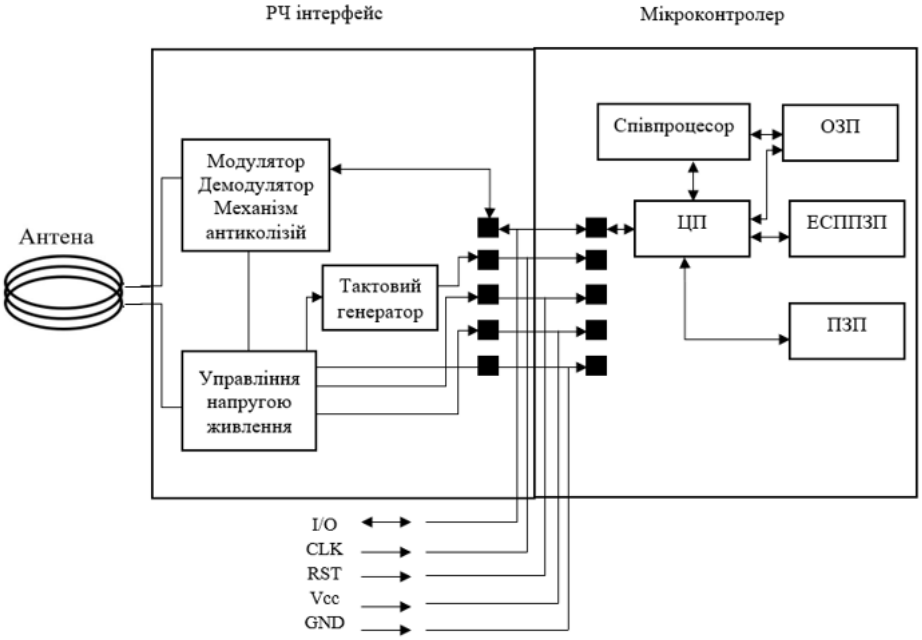

Рис. 1.22. Архітектура біометричного паспорта із співпроцесором та подвійним інтерфейсом

Ядром чіпа в мікропроцесорній карті є центральний процесор (ЦП), який оточений чотирма функціональними блоками: співпроцесором, ПЗП, ЕСППЗП, ОЗП та портом вводу/виводу. Співпроцесор призначений для розвантаження ЦП від трудомістких операцій крипто захисту та інших обчислень біометричного паспорта. Як випливає з п. 1.1.1, ПЗП містить операційну систему чіпа, яка записується в ПЗП під час виготовлення, дані та програмний код містяться в ЕСППЗП під керуванням операційної системи. ОЗП є оперативною пам'яттю процесора, всі дані, що зберігаються в ній, губляться з вимкненням живлення чіпа.

Логічна структура даних, що зберігаються в електронному паспорті на базі SMART-карти, показана на рис. 1.23. Інформація логічно розбита на три рівні [25]:

1. загальна персональна іінформація;
2. список доступу;
3. біометрична інформація.

Три рівні логічної моделі зберігання даних задовольняють мікропроцесорні SMART-карти як з RFID, так і з подвійним інтерфейсом передачі даних.

Рівень 1

Персональні дані		
Фамілія по батькові		**Стать**
Дата народження	**Місце народження**	**Національність**
Номер паспорта	**Дата випуску**	**Строк дії**

Рівень 2

Віза 1	**Віза 2**	**Віза 3**		**Віза N**

Рівень 3

Біометричні дані

Рис. 1.23 Рівні логічної структури даних біометричного паспорта

Глибина рівня відповідає ступеню конфіденційності даних. На нижніх рівнях є найбільш захищені відомості.

Рівень 1 зберігає загальні відомості про власника картки, наприклад, ім'я, стать, дату народження, національність, номер паспорта, термін його дії тощо. Як правило, ця інформація дублюється на оболонці корпусу SMART-карти.

Ця інформація шифрується так, щоб до неї міг мати доступ лише авторизований хост-комп'ютер, і захищена PIN-кодом.

На рівні 2 зберігаються у цифровій формі у вигляді послідовних записів традиційні візи, що надаються цьому громадянину різними країнами. Інформація, що міститься у кожному запису, включає код країни, тип візи, дату випуску, термін дії, умови тощо [22]. Обсяг цього рівня досить великий, щоб дати користувачеві можливість мати одночасно кілька віз. Кожен запис про візу має бути незалежно зашифрований органом, що видав.

Рівень 3 містить особливо захищені відомості – персональну біометричну інформацію власника картки, що використовується для ідентифікації. Логіка мікропроцесора біометричного паспорта захищає від

редагування розділи SMART-карти, вміст рівня може бути обнулений після певної кількості неправомірних спроб доступу.

З урахуванням особливостей побудови файлової системи, біометричний паспорт містить ієрархічно організовану файлову систему. У корені ієрархії перебуває MF, кожен логічний рівень представлений одним DF. Елементом рівня є EF. Кількість файлів обмежена обсягом пам'яті SMART-картки. Як правило, біометричні паспорти мають об'єм пам'яті від 128 Кбайт до 512 Кбайт. Таким чином, мікро-чіпи біометричних пристроїв дозволяють модифікувати алгоритми обробки, зберігання та введення-виведення даних.

1.1.3 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації

Виходячи із відомостей, можна резюмувати список таких ознак, властивих кожному персональному ідентифікатору у складі МКПІ:

1. *Практичність,* полягає в невеликих розмірах, що тягне за собою низьке енергоспоживання і зручність повсякденного використання. Вбудоване джерело енергії відсутнє, достатню потужність для функціонування вбудованого кремнієвого CMOS-чіпа забезпечує виникнення напруги на відповідних контактах (SMART-карти) або електричний струм, вбудований в антені електромагнітним сигналом від зчитувача (пасивні RFID-мітки) [2].

2. *Безпека*. Виражена криптографічною складовою у вигляді мікропроцесорів, що реалізують алгоритми DES, RSA для безпечного обміну даними. Зокрема, у SIM-картах реалізується GSM-алгоритм A3A8 генерації ключа шифрування телефонних переговорів. Існують області пам'яті, захищені від запису або одночасного запису та читання. Персональні ідентифікатори мають власну операційну систему COS чи GSMFS, яка контролює доступ до файлів виходячи з рівня пріоритетності. Також застосовується метод аутентифікації у вигляді запиту PIN-коду. Зазвичай існує обмежена кількість спроб правильного введення (в основному не

більше 3 разів), після чого персональні ідентифікатори блокуються для використання.

3. *Доступність*. Немає необхідності мати спеціальні знання та навички, щоб використовувати функції МКПІ. Кінцевому користувачеві достатньо помістити їх у спеціальний хост-комп'ютер і виконати процедуру перевірки справжності.

4. *Мобільність*. МКПІ регулярно використовуються для комунікації, підтвердження своєї особистості та певних привілеїв, для фінансових операцій та просто у побуті. Вартість виробництва персональних ідентифікаторів дуже мала. Компанії, які виробляють їхню емісію для своїх клієнтів, не включають їхню вартість у вартість послуг та безкоштовно проводять процедуру ремісії у разі відмови.

5. *Уніфікованість*. Мікро-чіпи персональних ідентифікаторів залиті у пластмасовому корпусі типу STANDART. Зовнішній вигляд та розміри підкоряються міжнародним домовленостям. Від інших карт відрізняються лише пристрої типу PLUG IN. Вони мають розміри типу STANDART, однак у них навколо мікро-чіпа є перфорація спеціальної форми, що дозволяє при необхідності витягти мікро-чіп і вкласти у відповідний хост-комп'ютер.

6. *Обмеженість пам'яті*. Зважаючи на стандартизовані розміри, мобільність і зручність, пам'ять персональних ідентифікаторів складається з мінімуму комірок, здатних забезпечувати функціонування. У переважній більшості МКПІ загальний обсяг пам'яті персональних ідентифікаторів становлять ROM-, RAM- та EEPROM-пам'ять [2]. Вміст ROM-пам'яті створюється на етапі виробництва кристала та містить фундаментальні дані: операційну систему, шифрувальні алгоритми. RAM-пам'ять використовується для забезпечення достовірного функціонування алгоритмів, а також як буфер при обміні даними та тимчасового зберігання даних. Енергетично-незалежна область пам'яті, EEPROM призначена для запису спеціальної інформації та її довготривалого зберігання.

7. *Вразливість*. На сьогоднішній день жоден виробник МКПІ не гарантує надійності та довговічності зберігання інформації в системі. Зумовлено це багатьма причинами, що не залежать від дій власника. Одна з проблем мікро-чіпів персональних ідентифікаторів - виникнення в них можливої несправності. Пластикова карта, в яку вбудований чіп, досить гнеться і чим більше чіп, тим більша ймовірність його пошкодити. Незахищеність магнітної смуги, що містить дані, у разі її пошкодження може призвести до втрати доступу до портативної системи. Вплив ненавмисних (паразитних) та навмисних електромагнітних випромінювань та перешкод, наявність яких погіршує ефективність роботи ідентифікаторів, виникнення випадкових помилок запису, модифікації даних, що зберігаються, можуть незворотно вплинути на працездатність пристрою.

Очевидно, що тільки перші п'ять критеріїв є сильними сторонами портативних систем, чим і пояснюється їхнє швидке поширення та активне впровадження у інформаційно-телекомунікаційні системи [15]. Критерії обмеженості обсягів пам'яті та відсутність механізмів протидії виникненню випадкових помилок є вимушеним заходом виробників персональних ідентифікаторів та компромісним кроком на шляху до практичності, уніфікації та доступності. Незмінні розміри чіпа стримують масштабування підсистеми зберігання даних, дозволяючи розвиватися тільки шляхом удосконалення конструкції та елементів пристроїв, що запам'ятовують, а більшою мірою за рахунок вдосконалення методів і алгоритмів зберігання даних. Перші екземпляри портативних систем мали сотні біт пам'яті для зберігання даних. Розвиток елементної бази та технологічних процесів виготовлення напівпровідникових виробів призвів до появи у 2010-2012 роках пристроїв до сотні мегабайт пам'яті [6]. Таким чином, в даний час з'явилася можливість підвищення надійності зберігання даних. Цільове призначення МКПІ та їх орієнтація на забезпечення життєдіяльності людини в інформаційному співтоваристві обумовлює необхідність створення їх на

базі стійких до відмови пристроїв, зводячи до мінімуму появу можливих помилок або ненавмисної модифікації даних.

1.2 Аналіз помилок, виникаючих в мікрокомп'ютерних комплексах персональної ідентифікації

Існуючі підходи до зберігання інформації, потребують модернізації, оскільки часто не забезпечують належного ступеня надійності МКПІ, не містять механізмів, спрямованих на відновлення даних після збою.

Для визначення напрямів модифікації МКПІ необхідний диференційований аналіз апаратних та програмних особливостей аналізованих портативних систем. На основі проведеного аналізу побудовані критерії, які повинні задовольняти алгоритми обробки, зберігання та введення-виведення даних, націлені на застосування у різних МКПІ.

1.2.1 Можливості застосування підвищення надійності SMART-карт

Клас SMART-карт одна із найпоширеніших персональних ідентифікаторів, застосовуваних нині у МКПІ [38]. Найбільш простим видом SMART-карт є карти з мікросхемою пам'яті. Дані МКПІ інтегрують у собі енергонезалежну пам'ять та схему введення-виведення даних.

Як було зазначено, для зберігання ідентифікаційних даних у картах з мікросхемою пам'яті використовується постійний пристрій ПЗП. У пам'яті ПЗП можна зчитувати дані, але в ній заборонено запис даних. Для зберігання інших даних, у тому числі змінюваних, у SMART-картах використовується ЕСППЗП. Даний вид пам'яті також як і пам'ять ПЗП є енергонезалежною, але на відміну від неї в ЕССПЗП можна записувати в період експлуатації чіпа. Однак зазначимо, що як правило, цю пам'ять поділяють на області, дозволені лише для читання, чи дозволені для читання і запису.

Пам'ять ЕСППЗП є сторінково-орієнтованою із середнім розміром сторінки 1-32 байт. В силу технічних особливостей ЕСППЗП, для запису

даних у комірку пам'яті необхідно попереднє стирання всієї сторінки [5]. Це означає, що запис відбувається завжди для всієї сторінки. Швидкість запису в ЕСППЗП становить близько 3-10 мс на одну сторінку пам'яті. Об'єм пам'яті ЕСППЗП від 64 до кількох сотень кілобайт. Оскільки обсяг пам'яті ЕСППЗП SMART-карт обмежений, виникають певні умови для використання алгоритмів і методів завадостійкого кодування даних, що зберігаються на карті. Сторінково-орієнтований характер пам'яті ЕСППЗП також створює певну специфіку використання методів завадостійкого кодування в цьому виді пам'яті. Іншою технічною особливістю ЕСППЗП є обмеженість кількості процедур запису, які можна зробити на сторінку пам'яті ЕСППЗП. Ця межа сильно варіюється в залежності від конструктивних особливостей чіпа та таких умов експлуатації, як температура та напруга живлення чіпа, і знаходиться найчастіше в діапазоні 10000-100000 циклів запису/зчитування.

З метою безпеки даних, карти з мікросхемою пам'яті найчастіше забезпечуються також блоком безпеки, який керує доступом до ЕСППЗП, обмежуючи читання чи запис/стирання певних ділянок пам'яті. На рис. 1.24 схематично зображено блок підсистеми зберігання SMART-карти із мікросхемою пам'яті.

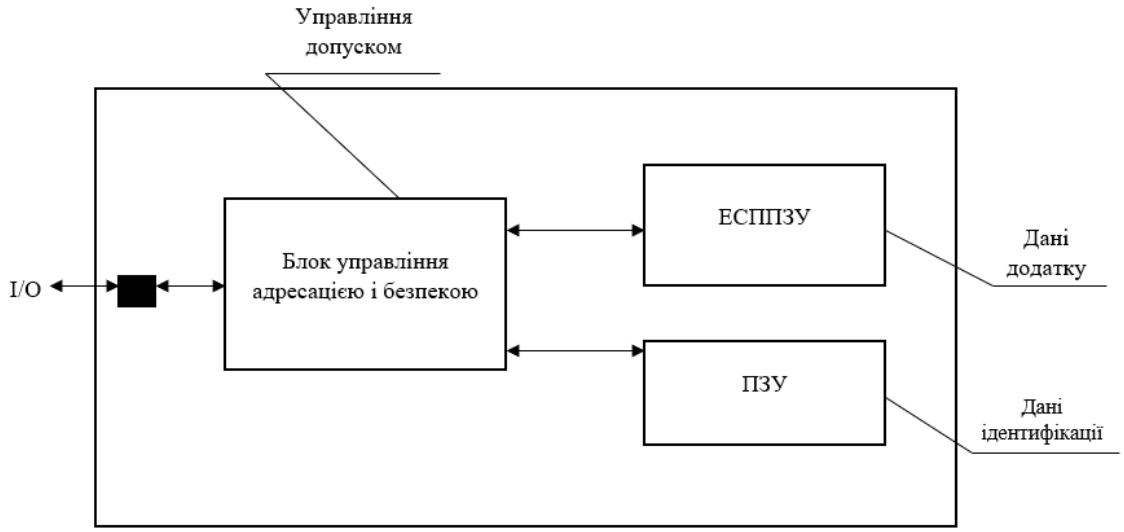

Рис. 1.24. Типова архітектура підсистеми зберігання SMART-карти з мікросхемою пам'яті

Важливою особливістю SMART-карт з мікросхемою пам'яті є те, що при обміні даними з хост-комп'ютером карта виконує лише низькорівневі команди, які їй передає пристрій для зчитування. Це команди адресації з читанням або записом за вказаними адресами пам'яті [9]. Більш сучасні карти цього типу також можуть виконувати просте шифрування даних, а також здійснювати автентифікацію користувача.

Більш складним, і водночас перспективнішим, видом SMART-карт є мікропроцесорні карти. У цих картах крім мікросхеми пам'яті є мікропроцесор (і співпроцесор). На рисунку 1.25 схематично зображено пристрій типової мікропроцесорної картки.

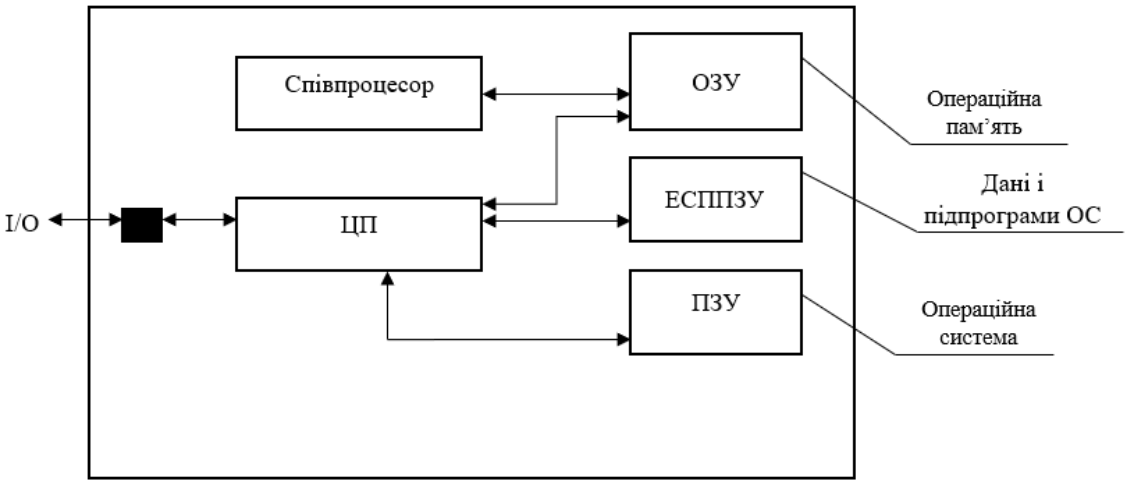

Рис. 1.25. Типова архітектура мікропроцесорної SMART-картки

У мікропроцесорних SMART-картах разом (іноді замість) з ЕСППЗП використовують технологічно інші види запам'ятовуючих пристроїв, що забезпечують енергонезалежну програмовану пам'ять. Таким є Flash-ЕСППЗП.

Flash-ЕСППЗП у технічному плані схоже на ЕСППЗП. Пам'ять цього типу також є сторінково-орієнтованою і має обмежену кількість циклів читання/запису. Розміри сторінок пам'яті Flash-ЕСППЗП зазвичай становить 64-128 байт.

Серед сучасних SMART-карт переважають карти двох типів: перші, в яких використовується і ПЗП та ЕСППЗП, другі, в яких ПЗП та ЕСППЗП замінює Flash-ЕСППЗП, на якій розміщені й операційна система та всі дані. Другий вид карт значно простіше для виробництва, тому що дозволяє вносити зміни в операційну систему картки, не змінюючи при цьому апаратну складову карти.

Підсумовуючи сказане вище, можна сказати, що мікропроцесори SMART-карт мають невеликі обчислювальні ресурси [10]. Тому одна з вимог алгоритмів та методів обробки, зберігання та введення-виведення даних, націлених на застосування в SMART-картах, це мала обчислювальна потужність, необхідна для них. Зауважимо, що наявність арифметичного співпроцесора дещо змінює ситуацію, так як обчислення, що підтримуються ним, можуть суттєво прискорити роботу деяких алгоритмів.

Будь-який файл (MF, DF, EF), записаний на SMART-карту, складається з заголовка файлу, що містить у собі інформацію про розміщення файлу та його структуру, і тіла файлу, що містить дані. В силу посторінкової орієнтації пам'яті ЕСППЗП та обмеженості числа циклів запису, заголовок файлу та тіло файлу розміщуються на різних сторінках пам'яті (заголовок файлу завжди незмінний). Структурованість елементарних файлів є важливою особливістю файлової системи SMART-карток, і будь-яка модифікація алгоритмів зберігання даних, що містяться в пам'яті картки, повинна зберігати структуру файлів. Зокрема, лінійний файл із записами фіксованої довжини повинен після кодування залишатися лінійним, а його записи мати однакову довжину [11]. Одним з природних способів вирішення цієї задачі є кодування кожного запису файлу окремо від інших.

При розгляді МКПІ зі SMART-картами, включаючи хост-комп'ютери (ATM, POS, ОКП та інші), виникає питання, які компоненти системи будуть відповідальними за виконання алгоритмів підвищення надійності.

Першим варіантом є здійснення роботи алгоритмів завадостійкого кодування та декодування у самій SMART-карті. Інший варіант, коли в

пам'яті SMART-карти лише зберігаються закодовані дані, а їхнє кодування та декодування відбувається засобами хост-комп'ютера.

З погляду безпеки перший варіант є кращим, так як менше число «зайвих» передач між картою та хост-комп'ютером. З іншого боку, у багатьох ситуаціях обробка даних на хост-комп'ютері не істотно впливає на безпеку даних.

Іншим критерієм під час виборів між цими варіантами є пропускна здатність каналу зв'язку. Пропускна здатність контактних SMART-карток становить близько 10000 біт/с, для безконтактних SMART-карток швидкість передачі даних ще нижче. При другому варіанті роботи кількість даних, що передаються, виявляється значно більше, що може надати істотну затримку в роботі системи. Крім того, при виявленні та виправленні помилки в переданих даних необхідна зворотна передача SMART-карті виправлених даних.

Сформулюємо групу умов, яким повинні задовольняти алгоритми та методи підвищення надійності, що застосовуються у SMART-картах:

1. З огляду на обмеженість пам'яті SMART-карт, надмірність структур, необхідні формування завадостійкість коду, повинні перевищувати 20% обсягу пам'яті.

2. Якщо кодування або перевірка інформації відбувається на SMART-карті, алгоритм кодування повинен успішно справлятися з роботою менш ніж за секунду за середньої потужності процесора 3.5 МГц і за наявності або відсутності арифметичного співпроцесора.

3. Виконуваний код алгоритмів завадостійкого кодування та декодування повинен займати у пам'яті SMART-карти не більше 10% обсягу пам'яті ЕСППЗУ.

4. Кодування має бути узгоджене зі сторінково-байтовою організованістю пам'яті. Наприклад, у разі блокування кодування блок повинен містити в собі цілу кількість сторінок, або, якщо сторінки пам'яті

великі, сторінка повинна містити ціле число блоків. Число додаткових коригувальних біт має бути кратною 8 для їхньої побайтової організації.

5. Кодування даних SMART-картки має зберігати структуру файлової системи, включаючи структуру файлів.

6. При кодуванні файлів SMART-карти, з метою безпеки, повинно зберігатися різне розміщення заголовків файлів та їх вмісту.

1.2.2 Можливості підвищення надійності магнітних карток

Пристрій магнітних карток значно простіше, ніж SMART-карток. Носієм пам'яті для цього виду карт є магнітна смуга, розташована у верхній частині карти. Відповідно до стандарту ISO 7811, магнітна смуга може містити до трьох треків. Перший і другий трек спеціалізуються лише для читання інформації, тоді як третій трек може використовуватись для запису інформації.

Відповідно до структур даних із п. 1.1.2:

- На першому треку може бути розміщено не більше 79 7-бітових символів;

- на другому треку може бути розміщено не більше 40 5-бітових символів;

- на третьому треку може бути розміщено трохи більше 107 5-бітних символів.

Кодування символів повністю описується стандартом ISO 7811. Усі символи старшого біту є бітом парності, тобто ведеться контроль цілісності інформації з допомогою перевірки на парність [12]. При цьому частина символів має спеціальне значення, такі як символи керування, мітки початку і кінця. Після позначки кінця ставиться останній символ – символ локального коду надлишковості, який обчислюється для даних, розміщених на треку.

Таким чином, для даних, що зберігаються на магнітній карті, здійснюється контроль цілісності за допомогою перевірки на парність і локальний код надлишковості. Істотним недоліком цих методів контролю і те, що де вони дозволяють обчислювати біти, у яких сталася помилка. Одним

із можливих способів підвищення надійності магнітних карт є оптимізація розміщення інформації на магнітній смузі та застосування алгоритмів корекції помилок. Даний спосіб здійснимо для карт, що застосовуються у певній сфері, але вимагає при цьому відповідної програмної підтримки хост-комп'ютерів, які працюють з картами [12]. Таким чином, підвищення надійності магнітних карток об'єктивно вимагає модифікації обробки, зберігання та введення-виведення даних не тільки персональних ідентифікаторів, але і всього МКПІ.

1.2.3 Можливості підвищення надійності в мікрокомп'ютерних комплексах на основі технології RFID

У МКПІ RFID-пристрої виконують функцію зберігання деяких ідентифікаційних даних та передачі їх хост-комп'ютеру за допомогою радіочастотного сигналу, при отриманні від нього відповідного сигналу.

Залежно від використовуваного носія даних, більшість RFID-міток становлять мітки з електронним носієм пам'яті на інтегральних мікросхемах. Другу групу систем радіочастотної ідентифікації складають мітки з носієм пам'яті, що базується на фізичних ефектах.

Практичний інтерес для реалізації алгоритмів та методів підвищення надійності шляхом модифікації програмної частини персональних ідентифікаторів представляє перша група міток, яка за своєю архітектурою схожа на безконтактні SMART-картки, і в свою чергу також як і SMART-картки ділиться на пристрої з електронною пам'яттю та мікропроцесором. Пам'ять даних базується на тих самих технологіях, що й пам'ять SMART-карт. Можливою принциповою відмінністю цих пристроїв від контактних SMART-карт є відсутність внутрішнього джерела живлення, що робить більш скрутним запис про ЕСППЗП.

RO-мітки з мікросхемою пам'яті містять лише ідентифікаційний номер. Як правило, в якості нього виступає порядковий номер мітки з приєднаною контрольною цифрою [14]. Виробниками мікросхем зазвичай гарантується унікальність цього номера.

Більш широкі можливості надають RW-мітки. Об'єм пам'яті даних міток становить від 16 байт до 64 Кбайт. Доступ хост-комп'ютера до таких міток часто здійснюється з допомогою блоків, тобто читання та запис інформації здійснюється єдиними блоками. Розміри блоку фіксовані, і зазвичай становлять 16 біт, 4 байти або 16 байт (але зустрічаються мітки з іншими розмірами блоків пам'яті). При цьому пам'ять ЕСППЗП міток залишається також орієнтованою на сторінку. Як правило, розміри блоків і сторінок вибираються так, щоб усі неперезаписані блоки даних розміщувалися в одній або двох сторінках пам'яті, а блоки, що перезаписуються, кожен в окремій сторінці. Таким чином, блоковий розподіл пам'яті є узгодженим зі сторінковою орієнтацією пам'яті ЕСППЗП, і в кожній сторінці розміщується ціла кількість блоків пам'яті.

Зазвичай кожен блок містить незалежну частину інформації, і природною є вимога, щоб будь-який алгоритм обробки та введення-виведення інформації зберігав початковий поділ інформації на блоки.

1.2.4 Можливості підвищення надійності біометричних паспортів

Біометричний паспорт є машино читаним носієм інформації з ідентифікаційними даними власника [39]. Усередині паспорта розташований чіп такий, як ті, що використовуються в мікропроцесорних SMART-картах, і радіопередавач для обміну даними з хост-комп'ютером [16]. Так само як і в SMART-картах, чіп має незалежну пам'ять ЕСППЗП, в якій розміщені ідентифікаційні дані власника, включаючи його біометричні характеристики.

Якщо говорити про українські біометричні паспорти, то на сьогоднішній день як біометричні дані використовується тільки фотографія власника паспорта. Ця фотографія зберігається у файлі формату jpeg і її розмір становить приблизно 1 Кбайт. Інші дані, що зберігаються в пам'яті ЕСППЗП - це дані, що надруковані на самому паспорті.

Розмір пам'яті ЕСППЗП в чіпах, які в українських паспортах, становить 32 Кбайт. Даного розміру пам'яті більш ніж достатньо для розміщення не тільки тих ідентифікаційних даних, що використовуються зараз, але і

багатьох інших додаткових біометричних характеристик, включаючи тривимірне зображення власника.

Пам'ять чіпів можна перезаписати, а також додати або виправити дані, які зберігаються в ній. Великий розмір пам'яті чіпа, що не використовується, надає великі можливості для застосування алгоритмів і методів підвищення надійності обробки та зберігання даних у пам'яті чіпа.

1.3 Аналіз помилок, які виникають у мікрокомп'ютерних комплексах персональної ідентифікації

Основою надійності будь-якої МКПІ є персональний ідентифікатор, оскільки вихід їх з ладу хост-комп'ютера не спричинить втрати персональних відомостей. Тому для аналізу помилок, що виникають у МКПІ, наводиться класифікація помилок, характерних для персональних ідентифікаторів.

По можливості усунення помилки можна класифікувати на непереборні та кориговані помилки:

- під непереборними помилками розуміються відмови блоків внаслідок їх дефектів, старіння, умов експлуатації;

- кориговані помилки - помилки, що виникають в елементах пам'яті без їх апаратної відмови. Причиною таких помилок є перешкоди в ланцюгах живлення, електромагнітні зовнішні перешкоди, температурна нестабільність.

За розподілом у пам'яті помилки можна розділити на рівномірно розподілені і кластерні (локалізовані) помилки:

- рівномірно розподілені помилки не мають певних місць зосередження;

- кластерні помилки мають «осередки появи». При виявленні помилки в елементі пам'яті, ймовірність виявлення помилок у прилеглих елементах пам'яті виявляється значно вище, ніж інших елементів. Прикладом таких помилок можуть бути подряпини на носії персональної інформації.

1.3.1 Помилки, характерні для магнітних карток

Магнітні карти діляться на карти з великою коерцитивною силою та малою коерцитивною силою. Дані в обох видах карток можуть бути пошкоджені під впливом магнітного поля [14]. Але важливою відмінністю між цими видами карт є сила такого магнітного поля. Для карток з високою коерцитивною силою, сила пошкодженого магнітного поля повинна бути значно вищою, ніж зустрічається в умовах експлуатації картки. Тому нині використовуються, зазвичай, карти першого виду.

Іншим видом шкідливих впливів на магнітні смуги є механічні ушкодження. Стандарт ISO 7811 визначає деякі параметри стійкості до механічних дій, яким повинні задовольняти магнітні карти. Тим не менш, дані види карток також мають свій термін служби і не є стійкими до будь-яких механічних впливів.

Таким чином, для магнітних карт характерним видом помилок є помилки, спричинені зношуванням магнітної смуги або іншим механічним пошкодженням магнітної смуги.

1.3.2 Помилки, характерні для мікрокомп’ютерних комплексів на основі технології RFID

Для різних типів RFID-міток характерні види помилок.

Якщо говорити про мітки з електронним носієм пам'яті, то помилками, що виникають при зберіганні даних, будуть помилки, характерні для ЕСППЗП. Залежно від типу міток, особливостей використання пам'яті ЕСППЗП та організації даних у ній, умов експлуатації буде виникати певна специфіка помилок, але в більшості випадків найбільш ймовірною помилкою буде залишатися збій сторінки пам'яті.

Робота однобітових міток полягає в різних фізичних ефектах. Тим не менш, для МКПІ цього типу буде справедливим поділ помилок на усувні та непереборні. Виникнення непереборної помилки означає повну непридатність мітки для подальшої експлуатації [17]. Усувна помилка може бути виправлена, але з великою ймовірністю свідчить про наявну апаратну

несправність пристрою, і тому в такому випадку мітка, як правило, також виводиться з експлуатації.

Зберігання даних у SAW-мітках у більшості випадків засноване на ефекті відбиття акустичних хвиль від рефлекторів, розташованих на п'єзокристалі. Виникнення помилки в даних, що зберігаються в такій мітці з найбільшою ймовірністю, означає механічне пошкодження рефлекторів або п'єзокристалу. Отже, даний вид помилки є непереборним, і SAW-мітка повинна бути виведена з експлуатації.

1.3.3 Помилки, характерні для біометричних паспортів

Дані біометричних паспортів зберігаються в пам'яті ЕСППЗП чіпа, і при зберіганні даних можливі самі помилки, що й для SMART-карт. Але розглядаючи специфіку використання ЕСППЗП у біометричних паспортах, можна зробити деякі зауваження.

По-перше, перезапис даних у біометричному паспорті є винятковою подією. У разі помилковості або не валідності даних, що містяться в паспорті, зазвичай змінюється весь паспорт [18]. Це означає, що кількість циклів запису в сторінках пам'яті ЕСППЗП не збільшується протягом усього періоду використання паспорта, а тому вихід з ладу сторінок пам'яті можливий тільки внаслідок шкідливих впливів і при закінченні терміну служби. Зауважимо, що заявлений термін служби паспорта встановлюється з урахуванням обмеженості терміну служби ЕСППЗП чіпа і становить 10 років.

По-друге, теоретично можливе додавання нових даних в ЕСППЗП в процесі експлуатації чіпа (наприклад, додаткові біометричні дані).

Таким чином, найбільш ймовірними причинами помилок зберігання даних у біометричних паспортах є шкідливі впливи. Також як і для SMART-карт, найбільш імовірною помилкою зберігання даних варто прийняти вихід з ладу сторінок пам'яті.

Висновки по першому розділу

Перший розділ дослідження присвячено аналізу особливостей функціонування та організації зберігання інформації в персональних ідентифікаторах МКПІ, а також можливостей підвищення надійності шляхом модифікації обробки, зберігання та введення-виведення даних та аналізу помилок, характерних для МКПІ, сформульовано постановку завдання дослідження. Проаналізовано інформацію про наявні мікрокомп'ютерні системи, які використовуються для особистої ідентифікації. Представлено архітектуру та особливості зберігання та розподілу інформації в RFID-мітках з функцією пам'яті. Визначено, що магнітні карти є найбільш схильними до різних помилок представниками класу МКПІ. На сьогоднішній день біометричні паспорти є найбільш складними та продуктивними представниками МКПІ.

Визначено, що критерії: практичність, безпека, доступність, мобільність, уніфікованість є сильними сторонами портативних систем, чим і пояснюється їх швидке поширення та активне впровадження у інформаційно-телекомунікаційні системи.

Представлено групу умов, яким повинні задовольняти алгоритми та методи підвищення надійності, що застосовуються у SMART-картах, Наведені можливості підвищення надійності магнітних карток. Проаналізовані помилки, які виникають у мікрокомп'ютерних комплексах персональної ідентифікації.

У результаті проведеного аналізу визначено, що методи та алгоритми обробки, зберігання та введення-виведення інформації, адаптовані для МКПІ, будуть стійкі до виникнення помилок у пам'яті персональних ідентифікаторів. Схеми завадостійкого кодування мають високу ефективність і стійкість до помилок. Представлені рекомендації щодо підвищення надійності МКПІ. Надані пропозиції щодо змін до базових стандартів для SMART-карт, магнітних карт, біометричних паспортів та RFID-систем, що регламентують обробку, зберігання та введення-виведення даних.

Проведений аналіз спрямований на отримання науково обґрунтованих технічних та технологічних рішень, спрямованих на вдосконалення методів та алгоритмів обробки, зберігання та введення-виведення даних у МКПІ.

РОЗДІЛ 2
РОЗРОБКА МЕТОДІВ І АЛГОРИТМІВ ОБРОБКИ, ЗБЕРІГАННЯ ТА ВВЕДЕННЯ-ВИВЕДЕННЯ ДАНИХ В МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСАХ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ

2.1. Схеми організації введення-виведення в комплексах персональної ідентифікації

У цьому розділі сформулюємо можливі варіанти модифікації введення-виведення інформації для підвищення надійності МКПІ. Результатом даного розділу є характеристика описаних варіантів функціонування.

Основними елементами будь-якої МКПІ крім персонального ідентифікатора та хост-комп'ютера, є канал зв'язку між ними.

Процедури завадостійкого кодування та декодування інформації, що зберігається у пам'яті МКПІ, можуть здійснюватися або хост комп'ютером, або безпосередньо персональним ідентифікатором. Крім того, у випадку SMART-карток можуть також виконуватися процедури шифрування та дешифрування даних, а також процедури завадостійкого кодування даних перед передачею їх по каналу зв'язку, та відповідне декодування.

Можливі дві основні схеми організації введення-виведення в МКПІ. Перша із них зображена на рисунку 2.1.

Рис. 2.1. Схема організації введення-виведення в мікрокомп'ютерному комплексі персональної ідентифікації у разі корекції на стороні хост-комп'ютера

За такої схеми функціонування, після читання закодованих даних, що зберігаються у пам'яті персонального ідентифікатора, відбувається їхнє декодування на хост-комп'ютері.

У разі виявлення помилки, дані повинні бути скориговані та знову відправлені до ідентифікатора для перезапису. Якщо персональний ідентифікатор передбачає тільки читання даних, без їх перезапису, схема спрощується шляхом видалення її процедур кодування та запису [28]. Але в цьому випадку виявлення помилки при декодуванні буде вказувати на необхідність заміни ідентифікатора.

Друга схема організації введення-виведення може бути реалізована тільки для персональних ідентифікаторів з чіпом (зокрема: SMART карт, мікропроцесорних RFID-міток та біометричних паспортів). Полягає в розміщенні всіх процедур стійкого до перешкод кодування та декодування даних у персональному ідентифікаторі.

Ця схема функціонування зображена на рис. 2.2.

Рис. 2.2. Схема організації введення-виведення у разі корекції на стороні персонального ідентифікатора

Оскільки така схема функціонування можлива лише для пристроїв з чіпом, в неї повинні бути поміщені процедури шифрування та дешифрування переданих даних, а також процедури кодування даних перед передачею по каналу зв'язку та декодування. Отже, друга схема насправді матиме вигляд, показаний рис. 2.3.

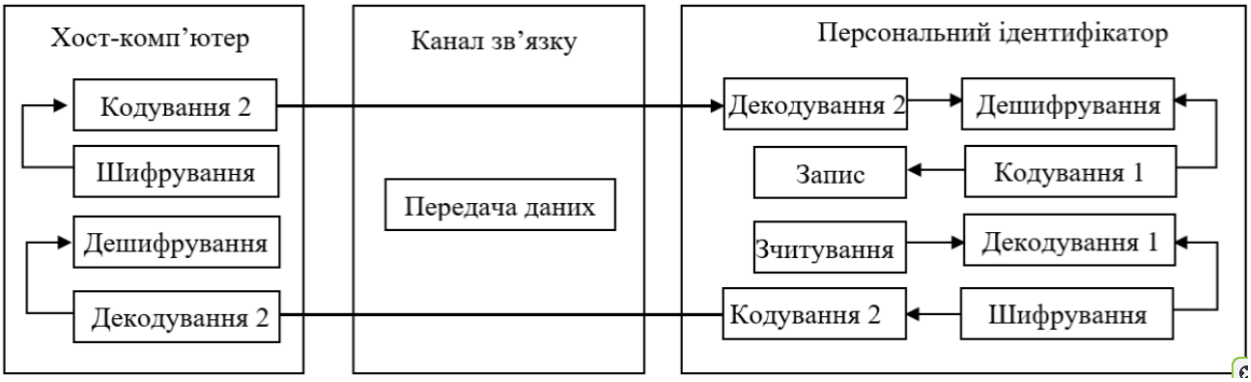

Рис. 2.3. Схема організації введення-виводу, ускладненої шифруванням у разі корекції на стороні персонального ідентифікатора

Кодування 2 та декодування 2 позначають процедури завадостійкого кодування, що застосовуються при передачі даних. Як такі можуть виступати як прості контрольні суми (наприклад, перевірка на парність), і різні циклічні коди. Кодування 1 та декодування 1 позначають процедури, що застосовуються при зберіганні даних у пам'яті персонального ідентифікатора.

Проведемо характеристику позначених схем організації введення-виведення за такими критеріями:

- забезпечення ефективності функціонування системи;
- забезпечення безпеки даних;
- забезпечення цілісності даних.

Розглянемо першу схему функціонування.

Допустимо, що хост-комп'ютер має достатню обчислювальну потужністю для забезпечення кодування/декодування даних за прийнятний час. Тому більшою мірою ефективність роботи системи залежить від каналу зв'язку та кількості переданих по ньому даних [28]. Для того, щоб хост-комп'ютер мав можливість здійснювати кодування чи декодування даних, йому має бути передано цілий блок даних. Виникає ситуація, коли для перевірки або кодування деякого набору даних хост-комп'ютер змушений запитувати у персонального ідентифікатора ширший набір даних. Ця обставина накладає суттєве обмеження на розбиття даних, що зберігаються в пам'яті персонального ідентифікатора, блоки кодування. З метою уникнути передачі надлишкових даних по каналу зв'язку, кожен блок даних по

можливості повинен вміщати тільки дані, що використовуються завжди разом. Іншим недоліком даної організації введення-виведення є ситуація, коли на код, що передається каналом зв'язку від персонального ідентифікатора до хост-комп'ютера, накладається помилка під час передачі [24]. Хост не зможе відрізнити її від помилки зберігання даних, і якщо зможе виправити її, в персональному ідентифікаторі буде здійснено перезапис даних на вихідні. У разі непоправності помилки буде зафіксована уявна невиправна втрата даних у пам'яті ідентифікатора при їх дійсній безпеці.

Крім того, ефективність функціонування МКПІ залежить також від обсягу надлишкової інформації. Пам'ять персонального ідентифікатора сильно обмежена, і тому має використовуватися максимально ефективно.

Зміст в одному кодовому блоці даних, які не завжди передаються разом, призводить не лише до зайвої завантаженості каналу зв'язку, а й до потенційного зниження рівня безпеки. Таким чином, міркування безпеки є додатковим аргументом на користь поділу даних при кодуванні максимально незалежних блоків.

В силу припущень, хост-комп'ютер забезпечує достатню обчислювальну потужність для роботи будь-яких алгоритмів завадостійкого кодування та декодування. Тому вибір методів завадостійкого кодування обмежується лише обсягом пам'яті персонального ідентифікатора, який можна витратити на надмірну інформацію коду. Для конкретних реалізацій МКПІ можуть бути підібрані індивідуальні ефективні методи обробки та зберігання даних.

Приведемо характеристику другої схеми організації введення-виведення МКПІ.

При даній організації введення-виводу каналом зв'язку МКПІ передаються самі дані, що передавалися без реалізації методів завадостійкого кодування даних, що зберігаються. Ефективність МКПІ повністю залежить від реалізації процедур кодування та декодування у логіні персонального ідентифікатора. Дані пристрої мають обмежену обчислювальну потужність,

тому важливим є побудова процедур кодування та декодування таким чином, щоб час, що витрачається на них, був мінімальний і невідчутний для власника (не більше 1 секунди).

Іншим обмеженням є внутрішня пам'ять персонального ідентифікатора. По-перше, при програмній реалізації методів кодування-декодування код займає певний обсяг пам'яті, який не повинен бути надто великим (не більше кількох Кб). По-друге, надлишкова інформація перешкодостійкого коду також як і для першого варіанту організації введення-виведення може займати значний обсяг пам'яті персонального ідентифікатора. Цей обсяг не повинен перевищувати меж, що залежать від конкретної сфери застосування портативного комплексу та його апаратних можливостей.

З точки зору безпеки, дана організація введення-виводу є досить надійною. Усі процедури завадостійкого кодування та декодування збережених даних реалізуються в межах ідентифікатора, тим самим залишаючи рівень безпеки МКПІ на колишньому рівні.

На відміну від першої схеми організації введення-виведення, крім обмеження на обсяг надлишкової інформації також накладаються обмеження на обчислювальну потужність, потрібну алгоритмам кодування та декодування [39]. Це суттєво обмежує можливості застосування завадостійкого кодування даних у МКПІ. Таким чином, розробка ефективних методів зберігання та обробки даних для цієї схеми значно складніше.

Підсумовуючи проведений аналіз, розуміємо, що для SMART карт, RFID-пристроїв з чіпом та біометричних паспортів найбільше придатною є схема організації введення-виведення, коли процедури кодування та декодування даних розміщуються в персональному ідентифікаторі.

Для магнітних карт єдиним можливим варіантом є розміщення процедур кодування та декодування на хост-комп'ютері.

2.2 Розробка методів і алгоритмів обробки, зберігання та введення-виведення даних в SMART-картах

Розділ присвячений аналізу методів та алгоритмів кодування даних у пам'яті SMART-карт, та можливостям застосування в даному типі персональних ідентифікаторів таких завадостійких кодів як код Хеммінга, код Ріда-Соломона та локальний код надлишковості LRC (Local Reconstruction Codes).

Результатами цього розділу є:

- Побудова методу зберігання даних, що дозволяє коригувати збої цілих сторінок пам'яті ЕСППЗП, а саме організація кодових блоків у пам'яті ЕСППЗП.

- Побудова моделі випадкового вибору сторінок для введення-виводу даних - однією з можливих, і в той же час, значною мірою узагальнюючої моделі організації роботи пам'яті ЕСППЗП. Її аналіз з точки зору застосовності перешкодостійких кодів, побудова ряду числових та функціональних показників ефективності застосування завадостійкого коду в даній моделі організації роботи пам'яті ЕСППЗП.

- Застосування кодів Хеммінгу, кодів Ріда-Соломона та коду LRC в моделі випадкового вибору сторінок для введення-виведення даних, модифікація алгоритмів зберігання даних на МКПІ зі SMART-картами.

Характерною помилкою, що виникає при зберіганні даних у пам'яті SMART-карти, є вихід із ладу сторінки пам'яті ЕСППЗП. Ця помилка може бути класифікована як пакетна непереборна помилка.

Тому основним завданням підвищення надійності для SMART-карток є забезпечення відновлення даних, що зберігалися в сторінці ЕСППЗП, та збереження їх в іншій робочій сторінці пам'яті [30]. Роботу процедур кодування/декодування можна вважати ефективною, якщо будуть відновлені дані хоча б за першого збою однієї сторінки пам'яті.

Надалі розглядатимемо лише ситуацію, коли з ладу виходить одна сторінка пам'яті.

Відповідно до границі Сінглтона, для того, щоб код мав здатністю відновлення однієї сторінки пам'яті, необхідно не менше двох сторінок із надмірною інформацією.

Одним із методів зберігання даних для виправлення пакетних помилок є перемежування коду. Даний метод може бути адаптований до нашої ситуації. А саме, як кодовані блоки даних можна розглядати послідовності із *i*-их символів кодованих сторінок пам'яті. Кожен символ представляє якусь частину даних фіксованої довжини. Зокрема, як такі можуть розглядатися біти або байти даних сторінки, або вся сторінка може розглядатися як символ. Відповідне розбиття на блоки зображено на рисунку 2.4:

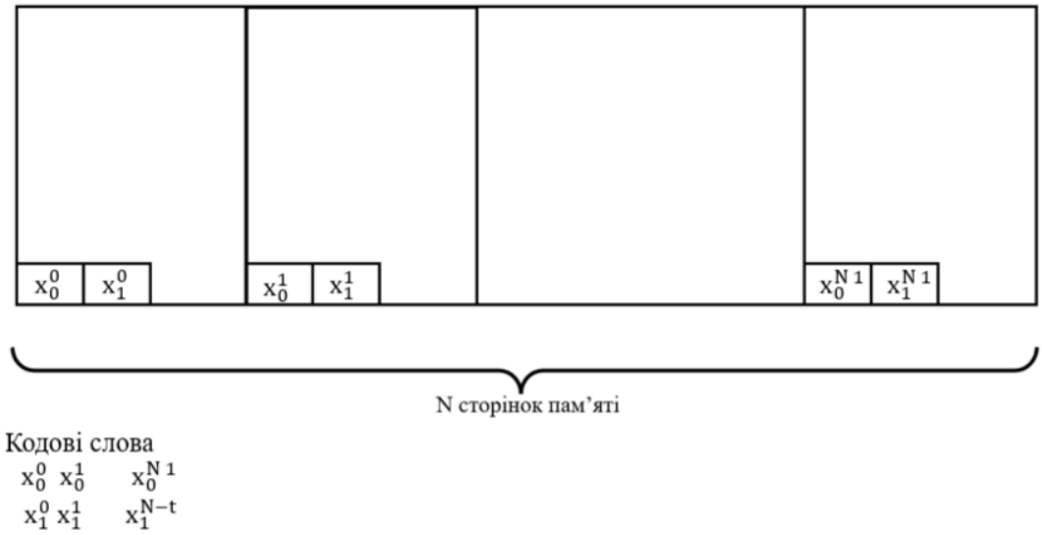

Рис. 2.4. Метод зберігання даних в програмованому запам'ятовуючому пристрої

Отримані кодові слова будуть містити не більше однієї помилки у разі виникнення збою однією зі сторінок. Тому для даних послідовностей даних достатньо буде використовувати код, що виправляє одну помилку. Прикладом таких кодів можуть бути коди Хеммінга та коди Ріда-Соломона. Але перш ніж аналізувати і порівнювати дані коди для цієї ситуації, проведемо загальний аналіз запропонованої методики кодування з метою

збереження структури файлової системи та файлів даних, найрозумнішим буде використання систематичних кодів. Іншими словами, для кодування даних необхідно резервувати якесь місце в пам'яті для надмірної інформації контролю, а частину пам'яті з даними залишати незмінною [31]. Будемо надалі називати сторінки частини пам'яті з надмірною інформацією сторінками контролю, а сторінки частини пам'яті з даними - називати сторінками даних.

Після перезапису будь-якої зі сторінок даних має відбуватися перерахунок контрольних символів коду та перезапис сторінок контролю. Логічним буде припустити, що в більшості кодів число сторінок контролю значно менше від кількості сторінок даних. Це означає, що число циклів стирання/запису для сторінок контролю значно швидше зростає, ніж сторінок даних. Найбільш імовірною є ситуація, що першою сторінкою пам'яті ЕСППЗУ, що дала збій, стане сторінка контролю.

Досліджуємо докладніше залежність числа циклів стирання/запису сторінок контролю від розподілу процедур запису між сторінками даних. Під процедурою запису даних розумітимемо процедуру запису деякої кількості сторінок даних, після якої відбувається завадостійке кодування даних та запис контрольної інформації у сторінки контролю.

2.2.1 Модель випадкового вибору сторінок пам'яті для введення-виведення даних

Одним із варіантів використання пам'яті ЕСППЗП є ситуація, коли запис може відбуватися багаторазово в різні сторінки пам'яті. При кожній процедурі введення-виведення (перезапису) даних перезаписується деякий набір сторінок даних залежно від дій, що здійснюються у МКПІ. Вважатимемо, що вибір таких сторінок для кожної процедури запису має випадковий характер і може бути передбачений тільки з якоюсь ймовірністю.

Як можна бачити, у такій моделі роботи пам'яті для опису процедур запису найбільш зручним та природним буде використовувати апарат теорії ймовірності.

Розглянемо перезапис i-ої сторінки даних як події, що має ймовірність P_i^d. Як один із варіантів, величину P_i^d можна визначити як відношення числа здійснених циклів стирання/запису i-ої сторінки до процедур запису у період експлуатації SMART-карти. Імовірності P_i^d разом становлять вектор ймовірностей $P^d = \left(P_0^d, \dots P_{k-1}^d\right)$, де – число k сторінок даних.

Події перезапису сторінок даних не можуть бути незалежними, так як при кожній процедурі запису перезаписується хоча б одна сторінка даних. Якби події були незалежними, існувала б ненульова ймовірність здійснення процедури запису без реального запису даних.

Щоб спростити модель роботи з даними у пам'яті, абстрагуємося від реальної організації роботи пам'яті та припустимо можливість здійснення "холостих" процедур запису [47]. Будемо припускати, що події перезапису сторінок даних можна розглядати як незалежні, а ймовірність холостої процедури запису вважати рівною $\prod_{i=0}^{k-i}\left(1 - P_i^d\right)$.

Для кожної сторінки даних із номером i також можна визначити залежні сторінки контролю i-ої сторінки, які можуть змінитися внаслідок перезапису i-ої сторінки даних. Ці залежності визначаються - використовуваним кодом і характеризують максимальну кількість сторінок контролю, які потрібно перезаписати внаслідок зміни i-ої сторінки даних.

Набір залежностей між сторінками даних та сторінками контролю можна описати за допомогою матриці Ω розмірності $k \times m$, де $m = N - k$ число сторінок контролю. Елемент Ω_{ij} дорівнює 1 за наявності відповідної залежності, і 0 інакше. Зауважимо, щоматриця Ω легко будується з урахуванням перевірочної матриці коду H. Матрицю залежностей Ω можна отримати, якщо вилучити з матриці H^1 всірядки крім тих, що відповідають контрольним символам коду, та замінити всі ненульові елементи на 1. Наприклад, перевірочна матриця H для коду Хеммінга (7,4) має вигляд:

$$H = \begin{pmatrix} 1110100 \\ 0111010 \\ 1101011 \end{pmatrix}.$$

Тоді матриця залежностей Ω між сторінками даних та контролю матиме вигляд:

$$\Omega = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

Введемо також величини γ_i^d і γ_j^c де $0 \le i \le k-1$, $0 \le j \le m-1$. Величина γ_i^d буде позначати кількість сторінок контролю, що залежать від i-ої сторінки даних, а γ_j^c позначатиме кількість сторінок даних, від яких залежить j-та сторінка контролю. Наступне твердження дає обмеження на значення величин γ_i^d і γ_j^c.

Твердження 2.1. Якщо код має мінімальну відстань Хеммінгу d між кодовими словами, то для будь-яких і таких, що $0 \le i \le k-1$ виконані нерівності:

$$\gamma_i^d \ge d-1 \tag{2.1}$$

Для величин γ_j^c завжди виконані нерівності:

$$\gamma_j^c \ge 1 \tag{2.2}$$

$$\sum_{j=0}^{m-1} \gamma_j^c \ge k(d-1) \tag{2.3}$$

Доказ. Вага ненульових кодових слів коду з мінімальною відстанню Хеммінгу d дорівнює d. Це означає, що в кожному стовпці перевірочної матриці, що відповідає контрольному символу, число ненульових елементів не менше $d-1$. Отже, кількість одиниць у будь-якому i-му рядку матриці залежностей Ω буде більше або дорівнювати $d-1$. Але це число і є γ_i^d.

Нерівність (2.2) очевидна: кожна сторінка контролю повинна залежати хоча б від однієї сторінки даних. Сума в лівій частині нерівності, є не що інше як сума елементів матриці Ω. Те, що ця сума більша або дорівнює правій частині нерівності, що випливає з нерівності (2.1), якщо застосувати до неї операцію підсумовування за будь-якими i.

Очевидно, що для зниження числа циклів стирання/запису сторінок контролю, а тим самим і продовження їх терміну служби, бажаним мати код, у якому кількість залежностей мінімальна.

Визначимо вектор ймовірностей $P^c = (P_0^c, \dots, P_{m-1}^c)$, як вектор, елементами якого є ймовірності P_i^c того, що при процедурі запису даних відбувся перезапис хоча б однієї сторінки даних, від якої залежить i-та сторінка контролю. Події перезапису сторінок контролю на відміну від подій перезапису сторінок даних загалом на разі не можуть бути розглянуті як незалежні [52].

Також визначимо вектор $\tilde{P}^c = \left(\tilde{P}_0^c, \dots, \tilde{P}_{m-1}^c\right)$, елементами якого є ймовірності $\tilde{P}_i^c$ перезапису відповідних сторінок контролю.

Імовірності P_i^cз наближенням оцінюють ймовірності $\tilde{P}_i^c$, будучи їх верхніми оцінками. Ці ймовірності не рівні, оскільки зміна інформаційних символів кодового слова ще не означає зміни всіх його контрольних знаків. Зв'язок між векторами P^c може бути описана за допомогою величини δ - умовної ймовірності того, що при настанні події зміни залежних сторінок даних, i-та сторінка контролю також зміниться. Залежність між P^c та $\tilde{P}^c$описується формулою:

$$\tilde{P}^c = \delta P^c \tag{2.4}$$

Імовірність δ, як можна показати, залежить лише від розміру сторінок пам'яті [44]. Справді, для будь-якого такого коду кожен контрольний символ кодових слів задається лінійним співвідношенням від інформаційних символів. Інформаційні слова, для яких контрольний символ набуває конкретного значення, може бути визначено рішенням відповідного лінійного рівняння. Кількість одержуваних рішень не залежить від значення контрольного символу, а визначається лише довжиною кодових слів. Тому якщо прийняти, що ймовірність появи будь-яких інформаційних слів коду однакові, контрольні символи кодового слова також з рівною ймовірністю прийматимуть будь-яке із значень алфавіту кодування. Якщо потужність

кодованого алфавіту дорівнює q, а відповідне число кодових слів s, то $\delta = 1 - \left(\frac{1}{q}\right)$. Але як потужність алфавіту кодування q, так і число кодових слів s залежить від довжини l двійкових послідовностей-символів. Якщо L — розмір сторінки пам'яті в бітах, то:

$$q = 2^l,$$

$$n = \frac{L}{l}.$$

Отримуємо:

$$\delta = 1 - \left(\frac{1}{2^l}\right)^{\frac{L}{l}} = 1 - \frac{1}{2^L}.$$

Мінімальним розміром сторінки пам'яті є один байт, звичайними ж - 16, 32, 64 байти. Таким чином, у більшості випадків з великим ступенем точності вважатимуться $\delta \approx 1$.

Вектори P^c та P^d також пов'язані між собою певним чином. Щоб описати цей зв'язок, введемо такі позначення:

$$\ln(v_1, \dots, v_n)^{dcf} = (\ln v_1, \dots, \ln v_n), \tag{2.5}$$

Також будемо позначати вектор $\left(\underbrace{1,1,\dots,1}_{n}\right)$ як 1^n. Наступне твердження дає рівність, що зв'язує вектори ймовірностей P^c та P^d.

Твердження 2.2. Нехай код має матрицю залежностей **Ω**. *Тоді для будь-якого вектора ймовірностей* P^d *справедлива нерівність:*

$$\ln(1^m - P^c) = \ln(1^k - P^d)\Omega. \tag{2.6}$$

Доказ. Елементи вектора $1^k - P^d$ є ймовірностями того, що відповідні сторінки даних не зміняться під час процедури запис. Аналогічно, елементи вектора $1^m - P^c$є ймовірностями того, що не змінюється жодна зі сторінок даних, залежних від відповідної сторінки контролю. Оскільки вважаємо події перезапису сторінок даних незалежними, ймовірність того, що певний набір сторінок залишиться незмінним, дорівнює набору відповідних ймовірностей

$1 - P_l^d$. Іншими словами ймовірність $1 - P_l^c$ дорівнює добутку ймовірностей незмінності залежних із нею сторінок даних.

Для обчислення добутку ймовірностей зручно розглядати замість них їхні логарифми. Завдяки цьому набору ймовірностей можуть бути замінені у сумі логарифмів ймовірностей. Відповідні суми логарифмів можуть бути записані у вигляді рівності (2.6).

Для вивчення залежності між векторами P^c та P^d корисним є таке твердження:

Твердження 2.3. Нехай P^dта $\bar{P}^d$ два вектору ймовірностей для сторінок даних такі, що $P^d \leq \bar{P}^d$. Тоді для відповідних векторів ймовірності P^cта $\bar{P}^c$для сторінок контролю справедлива нерівність $P^c \leq \bar{P}^c$.

Доказ. Матриця залежностей Ω складається з одиниць та нулів. Тому при перемноженні з вектором, що складається з невід'ємних (непозитивних) чисел, знову виходитиме невід'ємний (непозитивний) вектор. Вектори $\ln(1^k - \bar{P}^d)$ та$\ln(1^k - P^d)$ є непозитивними. Через нерівність $P^d \leq \bar{P}^d$ вектор $\bar{P}^d$може бути представлений як сума P^d і невід'ємного вектора v:

$$\bar{P}^d = P^d + v.$$

Тоді, оскільки логарифм є монотонно зростаючою функцією, $\ln(1^k - \bar{P}^d) = \ln(1^k - P^d - v) \leq \ln(1^k - P^d)$.

Звідси отримуємо, що $\ln(1^k - \bar{P}^d) = \ln(1^k - P^d) + u$, де u-непозитивний вектор. Звідси знаходимо:

$$\ln(1^k - \bar{P}^d)\Omega = \ln(1^k - P^d)\Omega + u\Omega,$$

$$\ln(1^m - \bar{P}^d) = \ln(1^m - P^d) + u\Omega,$$

$$\ln(1^m - \bar{P}^c) \leq \ln(1^m - P^c),$$

$$\bar{P}^c \geq P^c.$$

Суму елементів вектору ймовірностей $\tilde{P}^c$можна інтерпретувати як математичне очікування E_c числа сторінок, що перезаписуються контрольно. Так само визначимо величину математичного очікування E_d числа записуваних сторінок даних:

$$E_c = \sum_{i=1}^{m-1} \tilde{P}_i^c = \delta \sum_{i=0}^{m-1} P_i^c, \quad (2.7)$$

$$E_d = \sum_{i=0}^{k-1} P_i^d. \quad (2.8)$$

Зауважимо, що величина E_d через наявність незавершених операцій запису може набувати значення менше 1. Більш точно, математичне очікування E_d через допущення незавершених операцій запису збільшується в $1 - \prod_{i=0}^{k-1}\left(1 - P_i^d\right)$ рази (тобто зменшується). Величина E_c також зменшується порівняно з реальним шляхом множення на коефіцієнт $1 - \prod_{i=0}^{k-1}\left(1 - P_i^d\right)$.

Визначимо для завадостійкого коду, що задається на сторінках пам'яті, функцію $R(P^d)$ від вектора ймовірності P^d.

$$R(P^d)^{def} = \frac{E_c(P^d)}{E_d(P^d)}. \quad (2.9)$$

Поведінка функції $R(P^d)$ багато в чому описує ефективність досліджуваного коду у плані використання частини пам'яті для контрольних даних [54]. Чим менші значення набуває функція, тим більше ефективним є код. З огляду на вищесказане і визначення (2.9) функція $R(P^d)$ не залежить від припущення з урахуванням незавершених операцій запису.

Іншим важливим показником ефективності коду є векторна величина

$$\rho = (\rho_0, \dots \rho_{m-1}) = \frac{1}{maxP_j^d} \tilde{P}^c \quad (2.10)$$

Так само як і $R(P^d)$ функція $\rho(P^d)$ не залежить від припущення з урахуванням незавершених операцій запису.

Ця величина характеризує те, як частіше відбувається перезапис кожної сторінки контролю, ніж навантаженої сторінки даних. Параметр ρ є дуже важливим показником. Оскільки число циклів стирання/запису сторінок пам'яті ЕСППЗП обмежено, рано чи пізно відбувається їх вихід з ладу, і величина ρ може охарактеризувати кількість зіпсованих сторінок контролю до першого виходу з ладу сторінки даних.

При досягненні гарантійного значення числа циклів стирання/запису для i-ої сторінки контролю число циклів перезапису для сторінок даних буде

трохи більше $1/\rho_i$. Якщо вважати, що при досягненні гарантійного числа циклів стирання/запису сторінка пам'яті виходить із ладу, то насправді число сторінок пам'яті, необхідних для зберігання контрольної інформації, що дорівнює сумі величин ρ_i, заокруглених до більшого цілого. Позначимо цю величину і наведемо формулу для її обчислення:

$$\widetilde{m} = -\sum_{i=0}^{m-1}[-\rho_i]. \tag{2.11}$$

Розрахуємо, скільки пам'яті потрібно для зберігання даних ε, і скільки пам'яті фактично використовується $\tilde{\varepsilon}$:

$$\varepsilon = \frac{m}{k}, \tag{2.12}$$

$$\tilde{\varepsilon} = \frac{\widetilde{m}}{k}. \tag{2.13}$$

Величини ε і $\tilde{\varepsilon}$ можуть показувати межі застосування коду для розглянутого випадку. При цьому величина $\tilde{\varepsilon}$ є функцією $\tilde{\varepsilon}$ від вектора ймовірностей P^d та показує дійсний обсяг пам'яті, необхідний для роботи коду до виявлення першого збою сторінок даних.

На основі отриманої моделі у наступних розділах будуть розглянуто застосування відомих завадостійких кодів, які доцільно використати.

2.2.2 Застосування коду Хеммінга для обробки та зберігання інформації у SMART-картах

Позначатимемо через q потужність алфавіту коду $C(N, m)$. Оскільки будь-який символ кодового слова є двійковою послідовність фіксованої довжини l очевидно, що $q = 2^l$. А так як доступ до пам'яті ЕСППЗП побайтовий, найбільш доцільним буде вибирати l так, щоб або в байті вміщалося ціле число символів-послідовностей, або в символі поміщалося ціле число байт і одночасно у сторінці містилося ціле число символів. У першому випадку l може бути рівним 1, 2, 4, 8. Як правило, розмір сторінки пам'яті ЕСППЗП дорівнює 2 деякою мірою, а тому і в другому випадку l буде ступенем двійки. Отже, довжина l двійкова послідовність символу повинна дорівнювати ступеню двійки.

Розглянемо код Хеммінга $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1}-m\right)$. Число N сторінок пам'яті ЕСППЗП, в яких розміщуватиметься код, у загальному випадку не буде збігатися з $\frac{q^m-1}{q-1}$. Але необхідний код C може бути побудований шляхом застосування до коду Хеммінга операції укорочення коду [69], що полягає у видаленні з коду кількох інформаційних символів. Захищеність коду C_H від помилок при цьому анітрохи не зменшується [43].

Для можливості отримання коду C з коду Хеммінга C_H з допомогою операції укорочення, параметр m коду Хеммінга повинен задовольняти нерівності:

$$\frac{q^m-1}{q-1} \geq N,$$

з якого може бути отримана нижня оцінка для числа сторінок контролю:

$$m \geq log_q(N(q-1)+1). \tag{2.14}$$

Щоб показати, що код $C(N, N-m)$ отримано з коду Хеммінга $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1}-m\right)$ будемо також позначати його як $C_H\ (N, N-m)$.

Як приклад, впишемо в таблицю можливі параметри коду $C_H\ (N, N-m)$ для сторінок пам'яті розміром 32 байти з мінімальними значеннями m і відносний витрати пам'яті ε для кожного коду. У якості параметра N будемо брати різні ступені двійки як один з варіантів, що найчастіше зустрічаються в ЕСППЗП (таблиця 2.1).

Як очевидно з таблиці 2.1, відносний витрата пам'яті кодів виду $C_H\ (N, N-m)$ не великий, і зменшується із збільшенням довжини N кодових слів та потужності q алфавіту кодування. Більш реалістичну оцінку витрати пам'яті дає дійсну відносну витрату пам'яті $\tilde{\varepsilon}$, для оцінки якої необхідно детальніше дослідження роботи коду. Знайдемо матрицю залежностей коду.

Таблиця 2.1

Деякі коди $C_H\ (N, N-m)$ для 32-бітових сторінок пам'яті ЕСППЗП та алфавіту кодування з двійкових послідовностей довжини l

Довжина коду N	Довжина послідовності символу l (Біт)	Потужність q алфавіту коду	Число контрольних символів m	Відносна витрата пам'яті ε
128	1	2	8	6.67%
128	2	4	5	4.06%
128	4	16	3	2.4%
128	8	256	2 (гр. сигналів)	1.59%
256	1	2	9	3.64%
256	2	4	5	1.99%
256	4	16	3	1.19%
256	8	256	2 (гр. сигналів)	0.79%
512	1	2	10	1.99%
512	2	4	6	1.19%
512	4	16	4	0.79%
512	8	256	3	0.59%
512	16	65536	2	0.39%
1024	1	2	11	1.09%
1024	2	4	6	0.59%
1024	4	16	4	0.39%
1024	8	256	3	0.29%
1024	16	65536	2	0.2%

Матриця залежностей Ω коду Хеммінга $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1} - m\right)$ бути побудована з перевірочної матриці коду Хеммінга. При цьому виходить, що матриця Ω складатиметься з рядків, що є будь-якими комбінаціями одиниць та нулів. При цьому виконані умови:

1. у кожному рядку матриці не менше двох одиниць;

2. будь-який рядок, що являє собою впорядкований набір одиниць і нулів, зустрічається в матриці рівно $(q-1)^{x-1}$ раз, де x - кількість одиниць.

Таким чином, всі сторінки даних для коду Хеммінга $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1} - m\right)$мають від 2 до m залежних сторінок контролю. При цьому число сторінок

даних, що мають x залежних сторінок контролю всього $(q-1)^{x-1}C_m^x$ де C_m^x число поєднань із m по x.

Кожна сторінка контролю має $q^{m-1}-1\sum_{x=2}^{m}(q-1)^{x-1}\,C_{m-1}^{x-1}$ залежних сторінок даних. У разі укороченого коду число γ_l^c залежних сторінок даних для кожної i-ої сторінки контролю буде менше. Очевидно, найбільш раціональним при операції укорочення код відкидатиме сторінки, що мають максимальну кількість залежних сторінок контролю [71]. Також будемо робити вибір сторінок, що видаляються так, щоб після їх видалення різниця між величинами γ_l^c не перевищувала 1 і сторінки з меншим значенням γ_l^c при нумерації сторінок даних йшли першими.

Таким чином, операція укорочення коду Хеммінга буде однозначно заданою, а запис $C_H\,(N, N-m)$ буде позначати цілком певний код, що отримується з коду $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1}-m\right)$.

При видаленні з коду $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1}-m\right)$ не більше $(q-1)^{m-1}$ сторінок даних, для кожної сторінки контролю кількість залежностей зменшиться на кількість віддалених сторінок. При видаленні всіх сторінок даних, які мають x залежних сторінок контролю, всі величини γ_l^c зменшаться на $\frac{x}{m}(q-1)^{x-1}C_m^x$. Значення величин задовольняють формулам (2.2) і (2.3), але можливе знаходження їх сильнішої оцінки. Наступне твердження дає нижню межу для значень γ_l^c коду $\mathrm{C}_H\,(N, N-m)$.

Твердження 2.4. Якщо код $\mathrm{C}_H\,(N, N-m)$ *виходить з коду Хеммінга* $C_H\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1}-m\right)$, *операцією укорочення по сторінках даних, що мають максимальні значення* γ_l^c, *то для величин* γ_j^c *виконані нерівності:*

$$\gamma_j^c \geq \left[\frac{2k}{m}\right], \tag{2.15}$$

$$\gamma_j^c \leq q^{m-1}-1. \tag{2.16}$$

Доказ**.** Загальна кількість залежностей між сторінками даних та контролю дійсності (2.1) не менше $2k$. Оскільки скорочення коду

здійснюється так, щоб різниця між величинами γ_j^c не перевищувала 1, всяка величина γ_j^c буде не менше $\left[\frac{2k}{m}\right]$.

Друга нерівність виходить із нерівності для коду Хеммінга. При укороченні коду кількість залежностей лише зменшиться, а тому нерівність залишиться справедливою [45].

Нерівність (2.15) може бути переписана як $\gamma_j^c \geq \left[\frac{2}{\varepsilon}\right]$. Величина ε не має перевищувати 20%. Отже, можна зробити висновок, що для будь-якого коду $C_H(N, N-m)$ повинні бути виконані нерівності:

$$\gamma_j^c \geq \left[{}^{2}\!/_{1/5}\right] = 10 \tag{2.17}$$

Можна показати, що для коду $C_H(N, N-m)$ величини γ_j^c можуть бути скільки завгодно зменшені шляхом збільшення параметра m вихідного коду Хеммінга. При цьому значення ε збільшуватиметься. Цікавішим є той факт, що у багатьох випадках при збільшенні параметра m величини γ_j^c зменшуються «швидше», ніж збільшується значення ε.

Твердження 2.5. Позначимо середні величини γ_j^c кодів $C_H(N, N-m-1)$ та $C_H(N, N-m)$ як $\gamma^c(m)$ та $\gamma^c(m+1)$, відповідно. Якщо код $C_H(N, N-m)$ є сторінки даних, що мають число залежностей $\gamma_j^d > 2$, то виконано нерівність

$$\gamma^c(m+1) < \frac{m}{m+1}\frac{N-m-1}{N-m}\gamma^c(m), \tag{2.18}$$

$$\gamma^c(m+1) = \frac{m}{m+1}\frac{N-m-1}{N-m}\gamma^c(m). \tag{2.19}$$

Доказ**.** Спочатку розглянемо випадок, коли для коду $C_H(N, N-m)$ величини $\gamma_j^d = 2$ для всіх *i*. Число сторінок даних, що мають тільки 2 залежні сторінки контролю, для коду $C_H(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1} - m)$ дорівнює $(q-1)\mathrm{C}_m^2$. Отже, для похідного коду $C_H(N, N-m)$ має бути виконана нерівність $N - m \leq (q-1)\mathrm{C}_m^2$. Але тоді тим більше виконано нерівність $N - m - 1 \leq (q-1)\mathrm{C}_{m+1}^2$, що означає те, що код $C_H(N, N-m-1)$ для всіх сторінок

даних $\gamma_j^d = 2$. Значення $\gamma^c(m)$ та $\gamma^c(m+1)$ можуть бути легко підраховані для цього випадку і дорівнювати $\frac{2(N-m)}{m}$ та $\frac{2(N-m-1)}{m+1}$ відповідно. Підстановкою цих значень у рівність (2.19) легко переконатися у його справедливості.

Якщо для коду $C_H(N, N-m)$ є сторінки даних $\gamma_j^d > 2$, загальна кількість сторінок даних може бути представлена як сума $N - m = \sum_{x=2}^{t}(q-1)^{x-1}C_m^x + f$ де t - деяке число менше m, а f - позитивний залишок не більше ніж $(q-1)^t C_m^{t+1}$ для коду $C_H(N, N-m-1)$ можливе аналогічне уявлення $N - m - 1 \sum_{x=2}^{t}(q-1)^{x-1}C_{m+1}^x + g$. Так як для будь-яких $m >$ x $>$ 2 виконано $C_m^x < C_{m+1}^x$, отримуємо, що або $s < t$, або $s = t$ та $g < f$. Зіставляючи перші $N - m - l$ сторінок даних коду $C_H(N, N-m)$ сторінкам даних коду $C_H(N, N-m-1)$ отримуємо, що число залежностей для сторінок даних другого коду менше числа залежностей відповідних сторінок даних першого коду. Крім того, у коді $C_H(N, N-m)$ остання сторінка даних, що має a > 2 залежностей, виявляється «зайвою». Тому загальна кількість залежностей коду $C_H(N, N-m-1)$ як мінімум на 3 менше від кількості залежностей коду $C_H(N, N-m)$. Отримуємо:

$$\sum_{j=0}^{m}\gamma_j^c(m+1) \leq \sum_{j=0}^{m-1}\gamma_j^c(m) - a,$$

де a - максимальна кількість залежностей для сторінок даних коду $C_H(N, N-m)$. Здійснимо перетворення з нерівністю:

$$(m+1)\gamma^c(m+1) \leq m\gamma^c(m) - a,$$

$$\gamma^c(m+1) \leq \frac{m}{m+1}\gamma^c(m) - \frac{a}{m+1}.$$

Перетворимо нерівність (2.18):

$$\gamma^c(m+1) < \frac{m}{m+1}\gamma^c(m) - \frac{m}{m+1}\frac{1}{N-m}\gamma^c(m).$$

Для доказу достатньо показати, що $\frac{m\gamma^c(m)}{(m+1)(N-m)} < \frac{a}{m+1}$. Зауважимо, що $\frac{m\gamma^c(m)}{N-1}$ не що інше як середня кількість залежностей сторінок даних коду

$C_H(N, N-m)$. Це число обов'язково менше, ніж a. Справедливість нерівності (2.18) доведено.

Для оцінки ефективності коду Хеммінга нам потрібно зробити оцінку значень вектора p^c при різних значеннях p^d. Будемо рахувати, що нумерація сторінок даних будується методом вичерпування: в якості i-тої сторінки даних вибирається сторінка з максимальним значенням ймовірності перезапису серед ще не пронумерованих сторінок даних [43]. За такої нумерації сторінки з найбільшою ймовірністю перезаписи матимуть найменшу кількість залежностей.

Нехай ${}_1P^d = (p_1, \dots, p_1)$ і ${}_2P^d = (p_2, \dots, p_2)$ вектори ймовірностей з ймовірностями p_1 та p_2 такими, що $p_1 \le p_2$. В силу твердження 2.3, якщо ${}_1P^d \le P^d \le {}_2P^d$, то для відповідних векторів ймовірностей для сторінок контролю виконано нерівність ${}_1P^c \le P^c \le {}_2P^c$. Це означає, що залежність P^c від P^d значною мірою може бути вивчена при дослідженні більш простого окремого випадку, коли всі ймовірності P_i^d дорівнюють деякому значенню p. Відповідно до формули (2.6) всі ймовірності P_i^c матимуть значення

$$1 - (1-p)^{\gamma_i^c} \tag{2.20}$$

За формулами (2.7), (2.8) та (2.9) знаходимо математичні очікування числа сторінок даних, що перезаписуються, і контролю, і функцію $R(p)$:

$$E_d = (N-m)p,$$

$$E_c = \delta\left(m - \sum_{i=0}^{m-1}(1-p)^{\gamma_i^c}\right) = \left(1 - \frac{1}{2^L}\right)\left(m - \sum_{i=0}^{m-1}(1-p)^{\gamma_i^c}\right),$$

$$R(p) = \left(1 - \frac{1}{2^L}\right)\frac{m - \sum_{i=0}^{m-1}(1-p)^{\gamma_i^c}}{(N-m)p},$$

де L – розмір сторінки пам'яті в бітах.

Побудуємо графіки функції $R(p)$ при різних параметрах, розглянутих у таблиці 2.1. Для цього попередньо знайдемо відсутні параметри γ_i^c для представлених у таблиці кодів.

Як приклад обчислимо γ_i^c для першого коду.

Двійковий код Хеммінга при числі контрольних символів $m = 8$ має довжину кодових слів $2^8 - 1 = 255$ символів. Таким чином, для отримання потрібного коду має бути застосована операція укорочення та видалено $255 - 128 = 127$ символів, що відповідають сторінкам даних.

Оскільки $(2-1)^8 C_8^8 + (2-1)^7 C_8^7 + (2-1)^6 C_8^6 + (2-1)^5 C_8^5 = 93 < 127$, може бути видалені всі сторінки даних, що мають понад 4 залежні сторінки контролю. Число сторінок даних, що мають чотири залежні сторінки контролю, дорівнює $(2-1)^4 C_8^4 = 70$, і із них необхідно видалити 34 сторінки даних. Видаляти такі сторінки будемо попарно, об'єднуючи в пари сторінки, що мають разом як залежні всі 8 сторінок контролю. Таке розбиття однозначне, і при видаленні кожної пари всі параметри γ_i^c зменшуються на одиницю. Звідси отримуємо, що для будь-якої сторінки контролю

$$\gamma_i^c = 2^7 - 1 - \left(\frac{8}{8}(2-1)^8 C_8^8 + \frac{7}{8}(2-1)^7 C_8^7 + \frac{6}{8}(2-1)^6 C_8^6 + \frac{5}{8}(2-1)^5 C_8^5\right) - 17 = 46.$$

У таблиці 2.2 наведено значення величин γ_i^c для кодів таблиці 2.1.

Введем умовні позначення функції $R(p)$ для кодів таблиць 2.1. Будем записувати функції як $R1(p), R2(p), R3(p), R4(p)$ для чотирьох кодів з N = 128, і $R5(p), R6(p), R7(p), R8(p)$ для чотирьох кодів з N = 256. На рис. 2.5 і 2.6 представлені графіки названих функцій:

Таблиця 2.2

Число залежностей γ_i^c сторінок контролю від сторінок даних для різних кодів Хеммінга.

Довжина коду N	Потужність q алфавіту коду	Число контрольних символів m	Залежність сторінок контролю γ_i^c
128	2	8	$\gamma^c = 46$
128	4	5	$\gamma_0^c = \gamma_1^c = \gamma_2^c = 69, \gamma_3^c = \gamma_4^c = 70$
128	16	3	$\gamma_0^c = 116, \gamma_1^c = \gamma_2^c = 117$
128	256	2	$\gamma^c = 126$
256	2	9	$\gamma_0^c = \cdots = \gamma_3^c = 93, \gamma_4^c = \cdots \gamma_8^c = 94$
256	4	5	$\gamma_0^c = 170, \gamma_1^c = \cdots \gamma_4^c = 171$
256	16	3	$\gamma^c = 238$
256	256	2	$\gamma^c = 254$
512	2	10	$\gamma_1^c = \cdots \gamma_4^c = 192$

512	4	6	$\gamma_0^c = \gamma_1^c = 292, \gamma_2^c = \cdots \gamma_5^c = 293$
512	16	4	$\gamma_0^c = \gamma_1^c = 358, \gamma_2^c = \gamma_3^c = 359$
512	256	3	$\gamma_0^c = \gamma_1^c = 339, \gamma_2^c = 340$
512	65536	2	$\gamma^c = 510$
1024	2	11	$\gamma_0^c = \cdots = \gamma_8^c = 157, \gamma_9^c = \gamma_{10}^c = 158$
1024	4	6	$\gamma_0^c = \gamma_1^c = 698, \gamma_2^c = \cdots = \gamma_5^c = 699$
1024	16	4	$\gamma^c = 750$
1024	256	3	$\gamma^c = 766$
1024	65536	2	$\gamma^c = 1022$

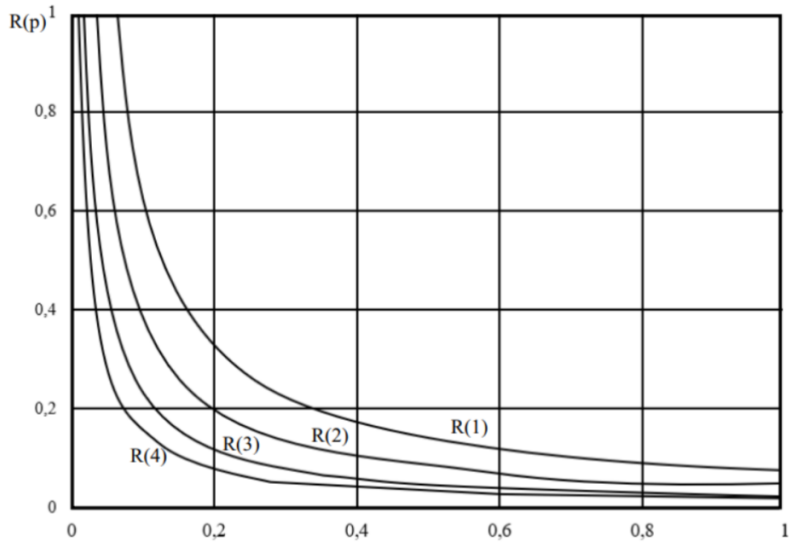

Рис. 2.5. Графіки функцій $R1(p), R2(p), R3(p), R4(p)$

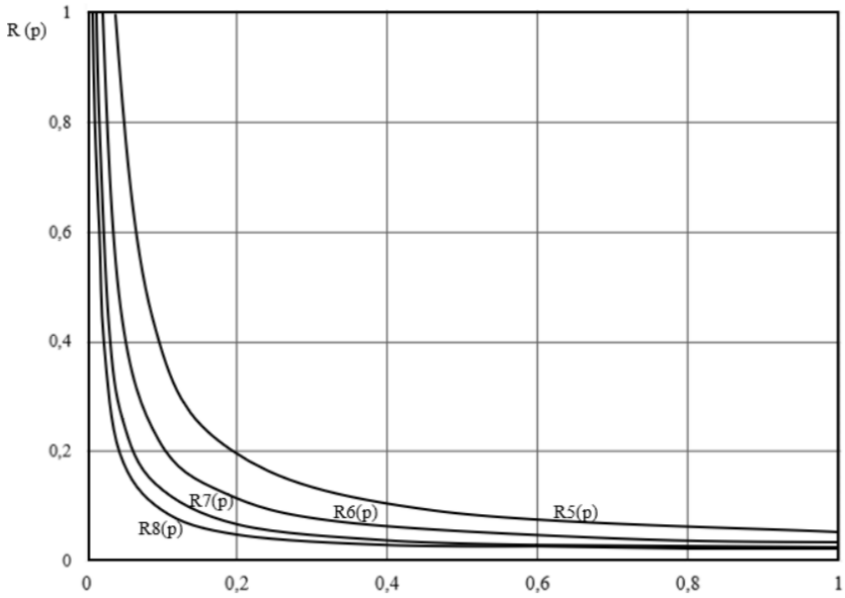

Рис. 2.6. Графіки функцій $1(p), R2(p), R3(p), R4(p)$

Більш точну інформацію про ефективність коду можна отримати шляхом вивчення векторної величини ρ (2.10) і дійсного числа сторінок контролю $\widetilde{m}$ (2.11), необхідного для забезпечення мінімальної цілісності сторінок контролю. На рисунку 2.7 наведено графіки функцій $\widetilde{m}(p)$ для кодів Хеммінга з таблиць 2.1 з довжиною кодових слів N = 128.

На рисунку 2.8 наведено графіки функцій $\widetilde{m}(p)$ для кодів Хеммінга з таблиць 2.1 з довжиною кодових слів N = 256.

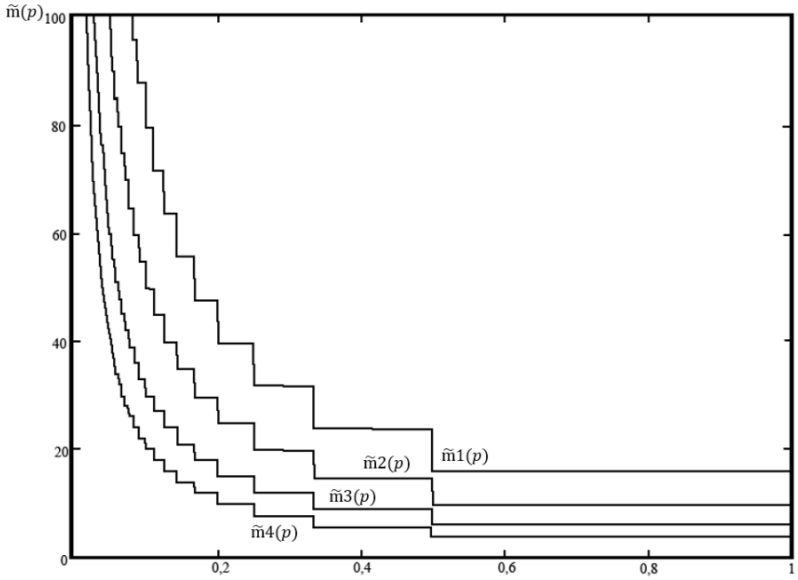

Рис. 2.7 Графіки функцій $\widetilde{m}(p)$ для кодів Хеммінга таблиця 2.1 с D'= 128

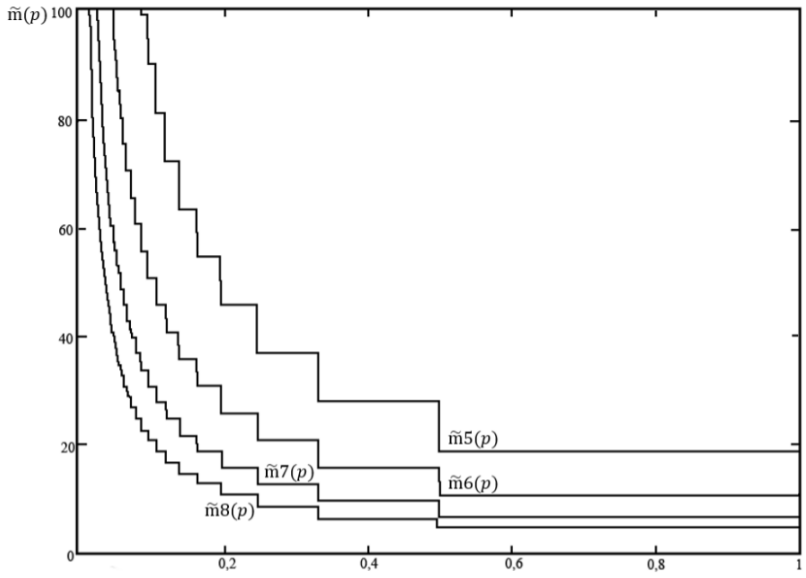

Рис.2.8 Графіки функцій $\widetilde{m}(p)$ для кодів Хеммінга таблиці 2.1 с N-256

На представлених графіках можна побачити, що зі зниженням ймовірності перезапису сторінок даних сильно підвищується потрібне число сторінок контролю. Це викликано тим, що при зменшенні ймовірності p математичне очікування чисельності перезаписаних сторінок даних може сильно знизитися, в той час як контроль сторінок таке зниження, якщо і буде, не помітить. Поведінка повних функцій $\widetilde{m}(p)$ досить близько до введенню гіперболи.

З графіків можна зробити висновок, що для розглянутих кодів допустима витрата пам’яті при кодуванні починається при ймовірності не менше 5-20%. Більше наглядно це можна побачити на графіках щодо витрат пам’яті $\tilde{\varepsilon}$, наведених на рисунку 2.9.

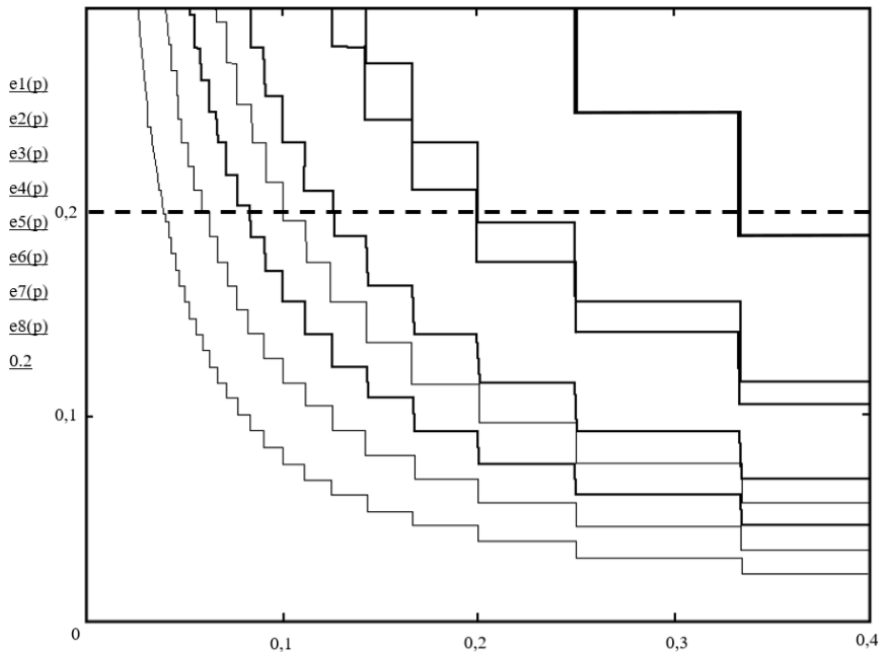

Рис. 2.9 Діюча відносна витрата пам'яті для кодів Хеммінга з таблиці 2.1 з N = 128 і N = 256

З графіків видно, що коди, що мають більшу довжину кодових слів, більш ефективно витрачають пам'ять. Пунктирною лінією на графіку позначений граничний рівень 20% витрати пам'яті, встановлені нами в попередніх розділах [64]. Навіть самий ефективний із кодів таблиці 2.1 має прийнятну витрату пам'яті, тільки початкова ймовірність перезапису сторінок

даних близько 5%, так як при умові, що при кожній процедурі запису в середньому буде перезаписано не менше 6 сторінок даних.

Розглянуті коди $C_H(N, N-m)$ є кодами з мінімальним значенням параметра m відповідно до формули (2.14). Відповідно до твердження 2.5 величини залежності γ_j^c зменшуються при збільшенні параметра m. Дані обставини можуть служити для використання більш ефективного коду з великим значенням параметра m. Як приклад, досліджуємо залежність ефективності коду від m для двійкових кодів виду $C_H(128,128-m)$. У таблиці 2.3 наведені значення величина γ_j^c для таких кодів.

Таблиця 2.3

Число залежностей γ_1^c сторінок контролю від сторінок даних для деяких двійкових кодів $C_H(128,128-m)$

Довжина коду N	Кількість контрольних символів m	Залежності сторінок контролю γ_1^c	Відносне використання пам'яті ε
128	8	$\gamma^c = 46$	6.25%
128	9	$\gamma_0^c = \gamma_1^c = \gamma_2^c = 35, \gamma_3^c = \cdots = \gamma_8^c = 36$	7.03%
128	10	$\gamma_0^c = 30, \gamma_1^c = \cdots = \gamma_9^c = 31$	7.81%
128	11	$\gamma_0^c = 26, \gamma_1^c = \cdots = \gamma_{10}^c = 27$	8.59%
128	12	$\gamma_0^c = \cdots = 23, \gamma_6^c = \cdots = \gamma_{11}^c = 24$	9.38%
128	13	$\gamma_0^c = \cdots = \gamma_5^c = 20, \gamma_6^c = \cdots = \gamma_{12}^c = 21$	10.16%
128	14	$\gamma_0^c = 17, \gamma_1^c = \cdots = \gamma_{13}^c = 18$	10.93%
128	15	$\gamma_0^c = \cdots = \gamma_5^c = 15, \gamma_6^c = \cdots = \gamma_{12}^c = 16$	11.71%
128	16	$\gamma_0^c = \cdots = \gamma_{11}^c = 14, \gamma_2^c = \cdots = \gamma_{15}^c = 15$	12.5%

Позначимо функції $R(p)$ для кодів $C_H(128,128-m)$ з таблиці 2.3 як $R_1(p), R_2(p), \dots, R_9(p)$. На рис. 2.10 зображені графіки заданих функцій в області, де їх поведінка представляє найбільший інтерес.

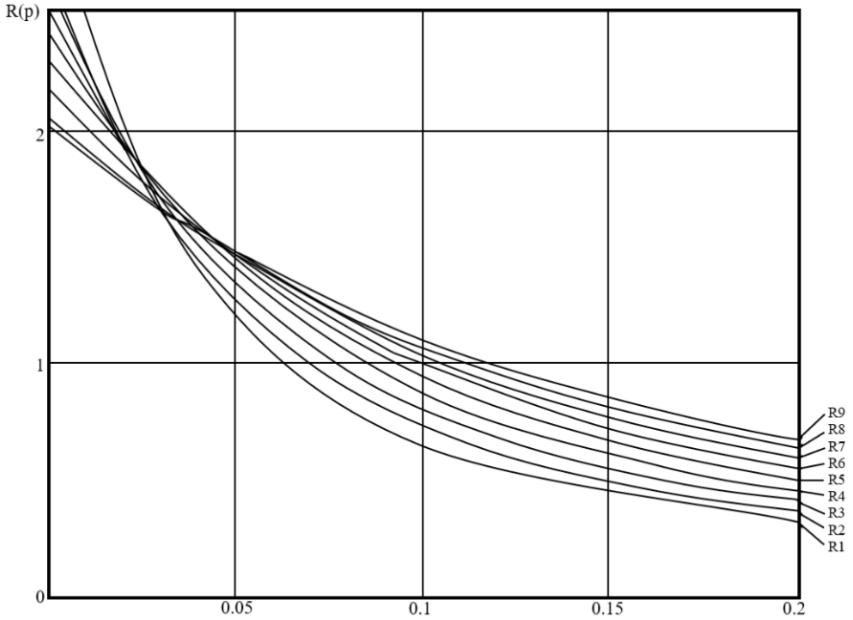

Рис. 2.10 Графіки функцій $R_1(p), R_2(p), \dots, R_9(p)$

Як можна побачити з представлених графіків функцій, при малих значеннях ймовірності p коди $C_H(128,128-m)$ з великим значенням параметрів p, виявляються більш ефективними. Але при ймовірності великих деяких значень ситуація кардинально міняється, - більш ефективно діють коди з меншим значенням параметра m.

Відповідно до формули (2.19), збільшення параметра m при наявності сторінок даних із числом залежностей більше 2 не приводить до істотної зміни функції $R(p)$ навколо точки $p=0$. Дійсно, нехай для фіксованого значення q функція $R_1(p)$ позначає функцію $R(p)$ для коду $C_H(N, N-m)$, а $R_2(p)$ позначає функцію $R(p)$ для коду $C_H(N, N-m-1)$, причому код $C_H(N, N-m)$ не має сторінок даних з числом залежностей більше 2. Функції $R_1(p)$ і $R_2(p)$ визначаються за формулою:

$$R(p) = \left(1 - \frac{1}{2^L}\right)\frac{m - \sum_{i=0}^{m-1}(1-p)^{\gamma_1^c}}{(N-m)p} \approx \left(1 - \frac{1}{2^L}\right)\frac{m\left(1-(1-p)^{\gamma^c}\right)}{(N-m)p} =$$

$$= \left(1 - \frac{1}{2^L}\right)\frac{m\left(1-\left(1-\gamma^c p + O(p^2)\right)\right)}{(N-m)p} = \left(1 - \frac{1}{2^L}\right)\frac{m\left(\gamma^c + O(p)\right)}{(N-m)p}.$$

Використовуючи формулу (2.19) здійснюємо перетворення різності функцій $R_2(p) - R_1(p)$:

$$R_2(p) - R_1(p) = \left(1 - \frac{1}{2^L}\right)\left(\frac{(m+1)\gamma^c(m+1)}{N-m-1} + \frac{(m+1)O_1(p)}{N-m-1} - \frac{m\gamma^c(m)}{N-m} - \frac{mO_2(p)}{N-m}\right) = = \left(1 - \frac{1}{2^L}\right)\left(\frac{m\gamma^c(m)}{N-m} + \frac{(m+1)O_1(p)}{N-m-1} - \frac{m\gamma^c(m)}{N-m} - \frac{mO_2(p)}{N-m}\right) = \left(1 - \frac{1}{2^L}\right)O'(p).$$

Таким чином, при достатньо малих значеннях p функції $R_1(p)$ і $R_2(p)$ є майже рівними.

При збільшенні параметра m $q-$значного коду $C_H(N, N-m)$ врешті-решт наступає момент, коли всі сторінки даних будуть мати рівно 2 залежності. Тому доречним буде ввести позначення M для мінімальних значень параметра m, при якому всі сторінки даних коду $C_H(N, N-m)$ мають рівно дві залежності. Параметр M може бути знайдено як мінімальне значення m, яке задовольняє кордон Сінглтона та наступної нерівності:

$$(q-1)C_m^2 \geq N - m$$

Здійснимо перетворення нерівності:

$$\frac{q-1}{2}m^2\frac{q-3}{2}m - N \geq 0,$$

$$\mathrm{m} \geq \frac{\mathrm{q}-3+\sqrt{(\mathrm{q}-3)^2+4\mathrm{N}(\mathrm{q}-1)}}{2(\mathrm{q}-1)}. \qquad (2.21)$$

Як видно з нерівності (2.21), параметр M залежить тільки від довжини коду N і потужності алфавіту кодування q. Більш того, можна зробити

висновок, що при достатньо великому значенні q при фіксованому N, параметр M буде дорівнювати 2. Найдем умову для q, при якому це буде так:

$$\frac{q-3+\sqrt{(q-3)^2+4N(q-1)}}{2(q-1)} \leq 2,$$

$$q-3+\sqrt{(q-3)^2+4N(q-1)} \leq 3q-1,$$

$$2q^2-Nq+N-2 \geq 0,$$

$$q \geq \frac{N}{2}-1.$$

Отримана умова того, що $M=2$, повинна бути узгоджена з нижньою оцінкою (2.14) для параметра m при заданих q і N. А саме, повинні виконуватися нерівності:

$$\log_4(N(q-1)+1) \leq 2,$$

$$N(q-1)+1 \leq q^2,$$

$$(q-1)(q-N+1) \geq 0,$$

$$q \geq N-1. \qquad (2.22)$$

З урахуванням нерівності $q \geq \frac{N}{2}-1$ отримуємо, що необхідною і достатньою умовою виконання рівності $M = 2$ є умовою (2.22).

Розглядаємо, як змінюється дійсна відносна витрата пам’яті для кодів $C_H(N, N-m)$ в залежності від величини $\tilde{\varepsilon}$. Оцінку витрат пам'яті відповідно до визначення (2.11), (2.13) і (2.20) можна рахувати рівною:

$$\tilde{\varepsilon} \approx \frac{M\left(1-(1-p)^{\gamma^c}\right)}{(N-M)p}$$

де γ^c - середнє число залежностей для сторін контролю, яке в силу визначення m і твердження 2.5 може бути визначено як:

$$\gamma^c = \frac{2(N-M)}{M} = \frac{2}{\varepsilon},$$

де ε - відносна витрата пам'яті для коду. Остаточно отримуємо наступну формулу:

$$\tilde{\varepsilon} \approx \frac{\varepsilon\left(1-(1-p)^{\frac{2}{\varepsilon}}\right)}{p}.$$

Отримана функція є змінною функцією від ε. У свою чергу ε змінювальна функція від параметра M. Наступне, при зменшенні параметра m дійсна відносна витрата пам'яті коду $C_H(N, N-m)$ зменшується.

Зменшення параметра M може бути досягнуто тільки збільшенням параметра коду q. Мінімальне значення q, при якому M стає рівним 2, тобто коли починає виконуватися умова (2.22) будемо позначати Q.

Ми показали, що при малих значеннях ймовірності є найбільш оптимальним за всіма показниками витрата пам'яті Q-значним кодом тому, що цей код є оптимальним для інших значень p, тому для цього коду досягається границя Синглтона, і тому відносна витрата пам'яті буде мінімальною [45].

Тепер проведемо аналіз алгоритмів кодування та декодування Q-значного коду $C_H(N, N-m)$.

Відповідно до структурної контрольної матриці q - код Хеммінга, перший контрольний символ будь-якого кодового слова є сумою всіх інформаційних символів. Другий контрольний символ також є сумою всіх інформаційних символів, але помножених на різні елементи поля *GF(q)*.

Наступна, перша сторінка контролю одержується сумуванням за модулем 2 всіх сторінок даних, що становить *L(N-3)* байтових операцій складання $\oplus$, де L - розмір сторінок пам'яті в байтах. Для вирахування другої сторони контролю також необхідно виробляти *L(N-3)* байтових операцій $\oplus$, але це повинно бути виконано деякою кількістю операцій множення сторінок даних на елементах із поля *GF(q)*.

Розглянемо детальніше операції множення в полі *GF(q)*. Кожен елемент поля може бути представлений як подвійне слово довжини *l* або як багаточленні ступені не більш *l*-1 над полем *GF(2)*. Визначимо відповідність між цими двома представленнями таким чином:

$$\underbrace{0\ldots01}_{l} \leftrightarrow x^0 = 1$$
$$\underbrace{0\ldots10}_{l} \leftrightarrow x$$
$$\vdots$$
$$\underbrace{0\ldots00}_{l} \leftrightarrow x^{l-1}$$

Таким чином, помноження елемента x^S $GF(q)$ означає побітовий здвиг частини його молодших бітів і більш складне перетворення частини старших бітів. Таким чином, операція множення в $GF(q)$ складається з кількох побайтових зрушень і операцій складання $\oplus$. Зі збільшенням q складність операції множення стрімко збільшується, що створює значущі розрахункові складності для застосування в мікропроцесорних картах.

Першим із способів утворення операції множення є створення в пам'яті ПЗП таблиці умноження елементів x^S. З урахуванням комутативності операції множення для такої таблиці потрібно близько $\frac{l^2}{2}\frac{l}{8} = \frac{l^3}{16}$ байт пам'яті. При l = 16, розмір таблиць буде складати всього 256 байт пам'яті, що є допустимою витратою пам'яті для більшості SMART-карт. При використанні таблиць перемноження двох елементів 11. ..1 і 11. ..1 зводиться до завершення l^2 операцій складання в поле, що рівносильно $\frac{l^3}{8}$ побайтових операцій $\oplus$.

Множення і ділення елементів кінцевого поля простіше всього здійснюється за допомогою іншого способу представлення елементів, а саме, представлення їх як ступеня примітивного елемента поля. Дані представлення можна розмістити в пам'яті ПЗП у вигляді списку. Список буде складатися з логарифмів елементів поля, налаштованих в порядку зростання (порядок елементів поля визначається як лексикографічний для їх

подвійного представлення). Розмір списку складає $q\frac{l}{8} = l\frac{2'}{16}$ байт. При l = 8 розмір списку буде розміщено 256 байт, але вже при 7 = 16 розмір списку буде 64 Кбайт, що не є прийнятним для SMART-карт.

Тим не менше, цей спосіб дозволяє максимально прискорити виконання операцій множення та відокремлення, виконання цих операцій рівносильними операціями складання по модулю $q - 1$, хоча і існуватиме тільки при параметрі l, який не перевищує 8.

При використанні таблиць множення розрахунок другої сторінки контролю буде означати не більше $\frac{8L}{l}\frac{l^3}{8}(N-3) + L(N-3) = L(N-3)(l^2 + 1)$ операції складання $\oplus$. Обчислення двох сторінок контролю в загальній складності буде проводитися до $L(N-3)(l^2+1)$ операціями $\oplus$.

При використанні списку логарифмів елементів поля, обчислення сторінки контролю буде займати всього $2L(N-3)$ операцій $\oplus$ і $\frac{8L}{l}(N-3)$ модульних операцій складання чисел розміром $\frac{l}{8}$ байт.

Друге спрощення операції множення може бути досягнуто при скороченні коду Хеммінга $C_H(N, N-m)$ до коду $C_H(N, N-m)$.

Так як всі сторінки даних при $m = 2$ залежать від усіх сторінок контролю, пріоритет при виборі видаленої сторінки даних може бути відданий будь-яким із них. Але кожній сторінці даних при цьому відповідають різні поля $GF(q)$, на які вони повинні множитися при кодуванні. Видаленні сторінки повинні бути сторінками даних, які відповідають елементу, помноженому на найбільшу величину. Такими є елементи $GF(q)$, що містять у своєму подвійному представленні найбільше число одиниць, або містять одиниці в перших розрядах. Точна оцінка складності множення є складною задачею, і залежить від q. Будем віддавати пріоритет видалення тим сторінкам даних, у яких відповідний елемент контрольної матриці має перший єдиний біт. В силу пристрою контрольної

матриці q-коду Хеммінга, це означає, що видаляться будуть сторінки даних, які мають самий перший номер.

Процедура кодування для коду Хеммінга може бути значно спрощена, якщо помітити, що при кодуванні вже раніше закодованого інформаційного блоку, досить лише участь нових даних. Для надходження нового значення контрольного символу для кодів при характеристиці поля, яке дорівнює 2, достатньо додати до контрольного символу поточного значення інформаційного символу з відповідним коефіцієнтом (при перезаписі сторінок даних) і додати нове значення з тим же коефіцієнтом. Зауважимо, що суттєво для інформаційного блоку має бути застосований повний алгоритм кодування, скорочений же буде використовуватися наприкінці. Обчислювальна трудомісткість скороченого алгоритму кодування залежить від кількості перезаписуваних сторінок даних [84]. Якщо їх кількість більше половини інформаційного блоку, то скорочений алгоритм кодування може бути навіть більш трудомістким, чим повний алгоритм кодування. В інших же випадках скорочений алгоритм кодування менш трудомісткий.

Алгоритм декодування коду Хеммінга в значній ступені повторює алгоритм кодування. Для обчислення кодової помилки досить спочатку обчислити символи контролю і скласти з тими, що є. Так як число синдромів дорівнює 2, якщо вони не дорівнюють обоє нулю, перша із S_1 кодова помилка дорівнює величині помилки, а величина — виявляється рівною $\frac{S_2}{S_1}$ позиції помилки (якщо рахувати з одиниці).

Зверніть увагу, що код Хеммінга та його алгоритм декодування мають наступну особливість: для з'ясування фактів наявності помилок достатньо вирахувати тільки першу контрольну сторінку [79]. Справді, при виникненні рівно однієї помилки в кодовому слові, його перший контрольний символ при пересуванні не співпаде із попереднім значенням.

У реальному режимі роботи декодування можна спростити, обмежившись обчисленням першої сторони контролю. Це становить лише

частину операцій декодування, але дозволяє виявити помилки в більшості випадків $L(N-3)$. Якщо ж помилка виявлена, то для її виправлення необхідно виконати повний процес декодування, включаючи обчислення другої сторони контролю.

2.2.3 Застосування коду Ріда-Соломона для обробки та зберігання інформації у SMART-картах

Коди Ріда-Соломона $C_{RS}(q-1, q-1-2t)$ на відміну від кодів Хеммінга здатні виправляти більше однієї помилки і завжди задовольняють межі Сінглтона. У разі кодів, що виправляють $t = 1$ помилок, також можна будувати кодові слова, беручи один символ з кожної сторінки пам'яті.

У разі довільного значення параметра t найдоцільніше побудова кодових слів таким чином, щоб з кожної сторінки даних бралося t символів, а $2t$ контрольних символів так само розміщувалися по двох сторінках контролю. Оскільки код виправляє t помилок, будь-який збій однієї сторінки пам'яті буде скориговано.

Подібне використання кодів, що виправляють більше однієї помилки, має незаперечну перевагу в плані виправлення помилок, не пов'язаних зі збоєм сторінок пам'яті. У разі виникнення більш ніж однієї помилки в межах одного кодового слова такий код зможе виявити її на відміну від коду, що виправляє одну помилку.

З метою збереження файлової структури даних, так само як і у випадку кодів Хеммінга, найбільш доцільним є застосування систематичного кодування за формулою [35]:

$$c(x) = i(x) - r(x) * x^k. \tag{2.23}$$

Для кодів даного типу також виділятимемо в пам'яті ЕСППЗП сторінки даних та сторінки контролю.

За будь-якого значення параметра t код Ріда-Соломона буде мати дві сторінки контролю. Причому при зміні будь-якого інформаційного символу

слідуватиме обов'язкова зміна всіх $2t$ контрольних символів, а відтак і обох сторінок контролю [74]. Таким чином, кожна сторінка контролю залежить від усіх сторінок даних - величини залежностей γ_0^c і γ_1^c мають значення $q-3$.

Так само як і у випадку кодів Хеммінга, довжина кодових слів може не співпадати з довжиною коду Ріда-Соломона, і код, що вимагається, може бути побудований шляхом операції скорочення. Отримуваний таким чином код позначатимемо $C_{RS}(N, N-m)$, де $m = 2t$. Похідний код $C_{RS}(N, N-m)$ вочевидь повинен задовольняти умові:

$$q - 1 \leq Nt. \tag{2.24}$$

Зауважимо, що як і для кодів Хеммінга, $q = 2'$, а параметр t повинен бути таким, щоб число t - символів однієї сторінки пам'яті ділилося на t. Оскільки розмір сторінки пам'яті дорівнює зазвичай ступеню двійки, параметр t може бути ступенем двійки.

Для кодів Ріда-Соломона також можливим є вивчення витрати пам'яті. Але на відміну від кодів Хеммінга, тут очікувана витрата пам'яті для всіх кодів буде одним і тим же, тому що кількість сторінок контролю завжди дорівнює двом, і вони завжди залежать від усіх сторінок даних. Більше того, коди Ріда-Соломона будуть витрачати пам'ять також як код $C_H(N, N-M)$ з попереднього розділу.

Найважливіша різниця між кодами Хеммінга і Ріда-Соломона виявляється у характеристиках алгоритмів кодування і декодування.

Для систематичних кодів Ріда-Соломона основну обчислювальну складність процедури кодування за формулою (2.23) становить розподіл полінома ступеня N на поліном, що породжує ступінь $2t$. Якщо розглядати звичайну процедуру поділу полінома, то для її здійснення має бути проведено $2t(N-2t+1)$ операцій складання, та $2t(N-2t+1)$ операцій множення в полі $GF(q)$.

Так само як і у разі кодів Хеммінгу прискорення операції множення у полі $GF(q)$ може бути досягнуто за допомогою створення таблиці множення.

У цьому операція множення буде зводитися лише до l^2 операцій додавання на полі $GF(q)$.

Таким чином, кодування одного кодового слова становитиме $\frac{1}{8}\left(2tl^2(N-2t+1)+2t(N-2t+1)\right)=\frac{lt}{4}(N-2t+1)(l^2+1)$ побайтових операцій складання $\oplus$. Так як кількість кодових слів дорівнює $\frac{8L}{lt}$, сумарна трудомісткість обчислень при процедурі кодування буде становити $2L(N-2t+1)(l^2+1)$ операцій складання $\oplus$.

При використанні списків логарифмів елементів поля сумарна трудомісткість процедури кодування становитиме $2L(N-2t+1)$ операцій складання $\oplus$ та $\frac{16L}{l}(N-2t+1)$ операцій складання чисел розміром $\frac{1}{8}$ - байт

по модулю $q-1$.

Також, як і для кодів Хеммінга, для кодів Ріда-Соломона аналогічним чином може бути організований скорочений алгоритм кодування, коли враховуються лише актуальні інформаційні символи.

Алгоритм декодування складається з кількох етапів. Перший етап, - обчислення компонента ознаки $S_i, (1 \le i \le 2t)$, полягає у обчисленні значень кодових поліномів у $2t$ коренях полінома, що породжує $g(x)$. Так як працюємо в полі характеристики 2, загалом для всіх кодових слів має бути обчислено $8\frac{L}{l}$ значень S_i для непарних значень i. Значення інших алгоритмів S_i для парних i обчислюються шляхом зведення у квадрат алгоритмів з відповідним індексом.

Обчислення значення кодового полінома ступеня $tN-1$ за допомогою схеми Горнера становить $tN-1$ операцію множення та tN операцій додавання у полі $GF(q)$. З використанням таблиці множення $\frac{1}{8}(l^2(tN-1)+tN)$ буде складати операції додавання, та в загальній складності трудомісткість першого етапу декодування складе $LNt(l^2+1)$ операцій додавання. При використанні списку логарифмів елементів поля

трудомісткість першого етапу становитиме операції додавання та $\frac{8LNt}{l}$ операції модульної арифметичної додавання чисел розміром $\frac{l}{8}$ байт за модулем $q-1$.

Другий етап алгоритму - обчислення коефіцієнтів полінома пошукових помилок Λ(x) може здійснюватися прямим методом Пітерсона-Горенстейна-Цирлера, або шляхом Берлекемпа-Мессі.

Пряме обчислення передбачає перебір ваги помилки від t до 1 доти, поки матриця з рівняння (2.25) стане невираженою, після чого слід звернутися до матриці:

$$\begin{pmatrix} S_1 & S_2 & \cdots & S_1 \\ S_2 & S_3 & \cdots & S_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_l & S_{l+1} & \cdots & S_{2l-1} \end{pmatrix} \begin{pmatrix} \Lambda_l \\ \Lambda_{l-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} -S_{l+1} \\ -S_{l+2} \\ \vdots \\ -S_{2l} \end{pmatrix}$$

Для знаходження зворотної матриці необхідне виділення в оперативній пам'яті $\frac{lt^2}{8}$ байт для її розміщення, а тому при великих значення параметра t розмір матриці може бути порівняний з розміром оперативної пам'яті SMART-карт.

Пошук ваги помилки та обіг матриці можуть бути поєднані шляхом застосування алгоритму Гауса для вирішення рівняння (2.25). При цьому буде здійснено не більше $\frac{t(t-1)(t+3)}{2}$ операцій додавання, $\frac{t(t-1)(t+1)}{2}$ операцій множення, $\frac{t(t+1)(t+2)}{2}$ операцій розподілу.

При використанні таблиці множення операція поділу може $\frac{l(l-1)(l+3)}{2}$

бути зведена до операцій додавання в полі, або, по-іншому, до $\frac{l^2(l-1)(l+3)}{16}$ байтові операції додавання. Таким чином, обчислення коефіцієнтів поліномів Λ(x) рівносильне виконанню не більше за $\frac{1}{16}\left((t+1)l\left(\frac{(t+2)}{2}(l^2-3)+(l^2+2)l\right)+t(t-1)(t+3)\right)$ байтових операцій. Сумарна

обчислювальна складність другого етапу $\frac{L}{2t}\left((t+1)l\left(\frac{(t+2)}{2}(l^2-3)+(l^2+2)l\right)+t(t-1)(t+3)\right)$ операцій .

Використання списку логарифмів дозволяє скоротити кількість операцій з виконання другого етапу до $\frac{(l-1)(l+3)}{2}L$ операції $\oplus$ і $\frac{4L}{l}\frac{(l+1)(l^2+2)}{t}$ операції арифметичної додавання чисел розміром $\frac{l}{8}$ байт за модулем $q-1$.

Оскільки характеристика поля $GF(q)$ дорівнює 2, алгоритм Берлекемпа-Мессі складатиметься з t ітерацій. При кожній i-ій ітерації має бути проведено не більше однієї операції розподілу, $2i$ операцій множення та $2i$ операцій додавання в полі $GF(q)$ при обчисленні наближення полінома Λ(x), і не більше $2i$ операцій множення та $2i$ Берлекемпа-Мессі буде операцій додавання при обчисленні нелінійності.

При використанні таблиць множення складність алгоритму повинна становити не більше $\frac{lt}{8}\left((t+1)(l^2+1)+\frac{l(l-1)(l+3)}{2}\right)$ операцій складання $\oplus$ для одного кодового слова, і не більше операцій $\oplus$ для другого етапу загалом.

При використанні списку логарифмів обчислення другого етапу декодування становитимуть не більше $L(t+1)$ операцій складання $\oplus$ та $\frac{8L}{l}(t+2)$ операцій модульного арифметичного складання.

Третій етап алгоритму - пошук помилок шляхом знаходження коріння полінома помилок. Пошук коренів здійснюється за допомогою методу Чені, що полягає в послідовному обчисленні значень полінома при підстановці в нього α', де α примітивний елемент поля, використаний при побудові коду. Обчислення кожного значення полінома зводяться не більше ніж до операцій множення і $t-1$ операції складання в полі. Оскільки декодуємо укорочений код Ріда-Соломона, повинні перебиратися не всі ступені елемента α, а лише від 0 до $tN-1$. При використанні таблиці множення даний метод займатиме

не більше $\frac{tNL}{8}(tl^2+t-1)$ байтових операцій додавання для кожного кодового слова, і не більше $LN(tl^2+t-1)$ операцій додавання для всього етапу. Застосування списків логарифмів знижує обчислювальну складність третього етапу до $LN(t-1)$ операцій додавання та $\frac{8LNt}{l}$ операцій арифметичного додавання по модулю $q-1$.

Четвертий етап - пошук значень помилок, може бути реалізований або прямим методом вирішення системи лінійних рівнянь, або за допомогою алгоритму Форні.

При прямому методі вирішення, всю обчислювальну складність становить звернення матриці розмірності не більше $t*t$, що рівносильно виконанню не більше $\frac{t(t-1)(t+3)}{2}$ операції додавання, $\frac{t(t-1)(t+1)}{2}$ операцій множення, $\frac{(t+1)(t+2)}{2}$ операцій розподілу. Таким чином, обчислювальна складність четвертого етапу становитиме за використання таблиць множення не більше $\frac{L}{2t}\left((t+1)l\left(\frac{(t+2)}{2}(l^2-3)+(l^2+2)l\right)+t(t-1)(t+3)\right)$ операцій, а при використання списку логарифмів - не більше $\frac{(t-1)(t+3)}{2}L$ операцій і

$\frac{4L}{l}\frac{(l+1)(l^2+2)}{t}$ - операцій арифметичного додавання чисел розміром $\frac{l}{8}$ байт за модулем $q-1$.

Алгоритм Форні складається з методу множення двох поліномів ступенів не більше t і $2t$, обчислення похідного полінома, знаходження значень двох поліномів. Все це становить трохи більше t операцій розподілу, $\frac{7t^2+5t-4}{2}$ операцій множення, $\frac{7t^2+3t-4}{2}$ операцій додавання в полі для одного кодового слова.

Обчислювальна складність четвертого етапу із застосуванням малого ритму Форні з використанням таблиці множення складе $\frac{L}{t}\left(t+\frac{7t^2+5t-4}{2}l^2+\frac{7t^2+3t-4}{2}l(l-1)(l+3)\right)$ операцій додавання, при застосуванні списку

логарифмів $\frac{L}{2t}(7t^2+3t-4)$ побайтових операцій додавання і $\frac{4L}{lt}(7t^2+7t-4)$ арифметичних операцій додавання модулю $q-1$.

Зауважимо, що в декодуванні обов'язковим є лише перший етап обчислення компонент синдрому. Якщо вони виявляються всі дорівнюють нулю, подальше декодування не проводиться.

Алгоритм рішення може бути спрощений у разі $t=1$. Оскільки декодування зводиться до вирішення системи нелінійних рівнянь (2.26), досить помітити, що у цьому випадку рівняння

$$S_1 = X_1Y_1 + \cdots + X_lY_l$$
$$\vdots$$
$$S_{2t} = X_1^{2t}Y_1 + \cdots + X_l^{2t}Y_l$$

набувають вигляду:

$$S_1 = Y_1X_1$$
$$S_2 = Y_1X_1^2$$

Тоді місцезнаходження та величина помилки можуть бути легко знайдені за формулами:

$$X_1 = \frac{S_2}{S_1}$$
$$X_2 = \frac{S_1^2}{S_2}$$

Таким чином, при $t=1$ другий, третій і четвертий етап рівносильні виконанню $\frac{16L}{l}$ операцій розподілу та $\frac{8L}{l}$ операції множення в полі $GF(q)$, що при використанні таблиць становитиме $Ll(l^2+3l-3)$ операцій, а при використанні списку логарифмів елементів $-\frac{24L}{l}$ операцій арифметичного складання по модулю $q-1$.

Незважаючи на спрощення процедури декодування коду Ріда-Соломона при $t=1$, процедура повного декодування коду Хеммінга все одно виявляється як простіша у реалізації, так і менш складною. При цьому витрата пам'яті та коригувальні властивості цих кодів загалом збігаються.

Єдина перевага кодів Ріда-Соломона над кодами Хеммінгу може бути отримана за рахунок збільшення значення параметра t.

З представленого аналізу алгоритму декодування можна зробити такі висновки:

1. Найбільш доцільним з точки зору швидкодії та витрати оперативної пам'яті є застосування на другому етапі декодування алгоритму Берлекемпа-Мессі, на четвертому етапі алгоритму Форні.

2. Складність декодування залежить лінійно від розмірів сторінки пам'яті L, кількості N сторінок пам'яті, що кодуються параметром t. Зауважимо, що складність методу кодування зі збільшенням параметра t зменшується на відміну процедури декодування. У разі використання таблиць множення складність декодування має кубічну залежність від параметра l, при використанні списків логарифмів елементів складність декодування практично не залежить від l.

3. Застосування списку логарифмів елементів поля забезпечує значне підвищення швидкості алгоритму порівняно з використанням таблиці множення, але створює велику витрату пам'яті ПЗП, і тому застосування даного способу оптимізації можливе лише за параметра $l \leq 8$.

4. Застосування кодів Ріда-Соломона доцільно лише за значеннями параметра $l > 1$.

Якщо при $l \leq 8$ процедури кодування та декодування можуть бути зроблені порівняно швидкими за рахунок застосування списків логарифмування, то вже за більших значень l їх обчислювальна складність сильно зростає. З іншого боку, довжина коду Ріда-Соломона при $l = 8$ обмежена 255 символами, а в реальній ситуації потрібно кодування більшої кількості сторінок пам'яті. Крім того, збільшення параметра t також вимагає збільшення довжини кодових слів за фіксованого числа сторінок пам'яті.

Прискорення виконання методу кодування та декодування можна досягти за рахунок апаратної реалізації найбільш складних методів.

Найбільшу частину складних обчислювальних операцій у методах кодування та декодування становлять операції множення у полі $GF(q)$. У значно меншій кількості зустрічаються операції поділу, але вони виявляються найскладнішими. Поле $GF(q)$ має характеристику 2, тому операції множення і розподілу повинні мати порівняно просту схемну реалізацію, і можуть бути реалізовані в мікроконтролерах SMART-карт в командах співпроцесора.

2.2.4 Застосування локального коду надлишковості для обробки та зберігання інформації в SMART-картах

У попередніх розділах визначено можливості виправлення збою сторінок пам'яті за допомогою кодів, що виявляють щонайменше одну помилку. Всі ці коди мають мінімальну вагу кодових слів $d \geq 3$. Коди з мінімальною вагою $d = 2$ здатні лише виявляти поодиноку помилку, але не виправляти її.

Розглянемо гіпотетичну ситуацію, коли апаратно-програмні можливості SMART-карти дозволяють виявляти збій, а також номер сторінки, що вийшла з ладу. Наявність такої додаткової інформації дозволяє використовувати код $d = 2$ як код, що виправляє одну помилку.

Найпростішим кодом із мінімальною вагою $d = 2$ є код із бітом парності. Що стосується організації кодових слів, використовуваної нами, цей код виявляється нічим іншим як локальним кодом надлишковості (LRC). Цей код матиме одну сторінку контролю, яка дорівнює сумі за модулем двох сторінок даних. Отже, поточна витрата пам'яті на контрольну інформацію для цього коду є меншою у два і більше разів, ніж для кодів з вагою $d \geq 3$.

У порівнянні з кодами з вагою $d = 3$, число циклів перезапису в сторінках контролю також знизиться приблизно вдвічі. Відповідно в таку ж кількість разів знизиться кількість m сторінок контролю, необхідних для забезпечення збереження даних до першого виходу з ладу сторінки даних.

Алгоритм кодування для цього коду є досить простим і компактним, і його машинний код може бути розміщений в пам'яті SMART-карти.

Алгоритм відновлення втрачених даних може бути реалізований таким чином:

1. Апаратно-програмними засобами SMART-карти виявляється збій *i*-ї сторінки даних.

2. Втрачена сторінка даних обчислюється як сума інших сторінок даних та сторінки контролю.

Оскільки збій сторінки пам'яті може бути виявлений без застосування методу декодування, вона здійснюється лише після виявлення помилки. Сам метод декодування обчислювальної складності збігається з методом кодування. Код її алгоритму також є досить компактним і може бути розміщений у пам'яті SMART-карти.

2.3 Розробка методів і алгоритмів обробки, зберігання та введення-виведення даних в мікрокомп'ютерних комплексах персональної ідентифікації на основі технології RFID

Розглянемо методи завадостійкого кодування для RFID-міток з електронним носієм на інтегральних мікросхемах. Для інших МКПІ можливості підвищення перешкодостійкості зберігання даних значно менші, i , швидше за все, припускають модифікацію також хост-комп'ютера.

Результатом даного розділу є запропоновані методи зниження енерговитрат у RFID-пристроях на виконання процедур кодування та декодування.

RFID-мітки з електронним носієм пам'яті багато в чому схожі на SMART-карти: у них також є сторінково-орієнтована пам'ять ЕСППЗП обмеженого розміру, і також найвірогіднішою помилкою залишається збій сторінки пам'яті.

Тим не менш, є важлива відмінність, яка полягає у відсутності безперебійного джерела живлення [82]. Тому будь-які методи обробки даних, які здійснюються на самому RFID-пристрої, повинні створювати мінімальну витрату енергії.

Таким чином, при виборі завадостійких кодів для RFID-міток з'являється додатковий критерій - методи кодування та декодування, у разі їх реалізації в самому МКПІ, повинні бути мінімально енерговитратними.

Припускатимемо, що витрати енергії RFID-мітки на виконання методу кодування-декодування безпосередньо залежать від часу його виконання. Зниження енерговитрат виконання методу може бути досягнуто такими методами:

1. Зменшення розмірів кодованих інформаційних блоків. Оскільки метод кодування та декодування будуть обробляти менші обсяги даних, їх час виконання зменшиться. Зауважимо, що при зменшенні розмірів блоків, що кодуються, також збільшиться витрата пам'яті на контрольну інформацію.

2. Перенесення або часткове перенесення методів кодування та декодування на хост-комп'ютер. Даний спосіб передбачає зниження рівня безпеки МКПІ за рахунок надмірності даних, що передаються.

3. Апаратна реалізація методу кодування та декодування. Недоліком цього є те, що апаратна реалізація алгоритмів обробки даних з найбільшою ймовірністю привиде до підвищення собівартості RFID-пристрою.

Можливості застосування цих методів сильно залежать від самого RFID-пристрою, вимог до МКПІ.

У разі RO-міток витрата пам'яті t і реальна витрата пам'яті m збігаються, оскільки перезапису сторінок даних не відбувається. Таким чином, реальна витрата пам'яті на завадостійкий код виявляється порівняно невеликою. Ця обставина дозволяє використовувати для кодування інформаційні блоки меншого розміру, підвищуючи цим ефективність виконання методу кодування та декодування.

У разі RO -міток мінімальні розміри кодованих інформаційних блоків можуть бути сильно обмежені.

Повне перенесення методу кодування та декодування на хост-комп'ютер веде до необхідності передачі великих блоків даних, що знову ж таки вимагає додаткової витрати енергії RFID-мітки, впливаючи також на безпеку системи [98]. Але для деяких кодів ця проблема певною мірою може бути вирішена: практично будь-який метод декодування може бути поділений на етап роботи з усім кодовим словом та на етап роботи безпосередньо з помилками. У разі лінійних кодів перший етап включає обчислення синдрому помилки, другий етап - пошук помилки синдромом.

Метод декодування може бути розподілена наступним чином: в персональному ідентифікаторі здійснюється перший етап декодування, далі хост-комп'ютеру передається запитані дані разом із проміжними даними процедури декодування (наприклад, значення синдромів).

Для кодів Хеммінга основну частину методу декодування становить перший етап, коли обчислюються значення контрольних сторінок. Але цей етап може бути скорочений, якщо при обчисленні контрольних сторінок використовувати ті дані, які не будуть передаватися хост-комп'ютеру. Вже на хост-комп'ютері значення контрольних сторінок буде дораховано.

Для кодів Ріда-Соломона перший етап полягатиме у обчисленні компонента синдрому і як і для кодів Хеммінга, досить зробити обчислення тільки з тими сторінками пам'яті, дані яких будуть передані хост-комп'ютеру.

З огляду на специфіки використання RFID-пристроїв часто є ситуація, коли перешкодостійкий код повинен забезпечувати відновлення спотворених даних, лише виявлення факту збою у роботі пристрою. Найпростішим кодом, що має дану властивість, є код LRC.

2.4 Розробка методів та алгоритмів обробки, зберігання та введення-виведення даних у біометричних паспортах

Результатами цього розділу є: побудована модель послідовного вибору сторінок для введення-виведення даних, що найбільш реально описує роботу з біометричними даними [83]. Її аналіз у частині застосування в ній завадостійкого кодування, побудова числової характеристики, що описує захищеність коду.

Застосування кодів Хеммінгу та кодів Ріда-Соломона в моделі послідовного вибору сторінок для сторінок для введення-виведення даних.

Біометричні паспорти загалом ідентичні безконтактним SMART-картам. У плані застосування до них методів завадостійкого кодування їм справедливі багато висновків, отримані для SMART-карт і RFID-пристрою. Тим не менш, біометричні паспорти є більш вузьким класом персональних ідентифікаторів, і, отже, для них можуть бути побудовані більш специфічні моделі роботи пам'яті, і точніше оцінені багато показників ефективності завадостійкого кодування.

Серед вимог, що висуваються до методів завадостійкого кодування для біометричних паспортів, можна виділити наступні.

Як було зазначено раніше, перезапис даних біометричного паспорта є винятковою подією. Тому можна вважати, що для цього виду МКПІ відбувається лише поповнення пам'яті ЕСППЗП новими даними без перезапису старих. Для опису цієї організації роботи з пам'яттю необхідно визначити модель послідовного вибору сторінок введення-виведення даних.

Розглянемо ситуацію використання пам'яті ЕСППЗП, коли введення-виведення (перезапис) сторінок даних не відбувається (або відбувається рідко), а має місце лише запис нових даних у вільні сторінки пам'яті.

Для такої моделі використання пам'яті кількість циклів стирання/запису сторінок даних фактично не зростає, і тому протягом усього

періоду експлуатації залишається незначною порівняно з гарантійним числом циклів стирання/запису.

У такій ситуації вихід з ладу сторінок пам'яті виникає з найбільшою ймовірністю або в результаті зовнішніх впливів, або при терміні закінчення служби мікроконтролера. А тому найімовірнішими стають виходи з експлуатації відразу кількох сторінок пам'яті. Таким чином, більшого значення набувають коди, що виправляють більше однієї помилки.

Число циклів стирання/запису сторінок контролю на відміну від сторінок даних зростатиме в міру додавання в пам'ять нових даних, і можна припустити, що зростатиме пропорційно кількості даних.

Якщо кількість процедур запису досить велика, то ймовірність виходу з ладу сторінок контролю стає значно більшою, ніж для сторінок даних.

З надмірним збільшенням числа циклів перезапису сторінок контролю можна впоратися, якщо розбити масив сторінок даних на кілька частин, і кожен набір сторінок кодувати окремо [85]. При цьому захищеність від багаторазових втрат сторінок даних зросте, але разом з нею зросте кількість необхідних сторінок контролю.

Зокрема, можна використовувати коди, що виправляють одну помилку, застосовуючи їх до невеликих масивів сторінок даних. Але при такому підході не коригуватимуться збої відразу двох сторінок даних в одному підмасиві. Тому такий варіант кодування є менш надійним, ніж кодування повністю всіх сторінок даних кодом, що виправляє кілька помилок. Позитивним ефектом розбиття масиву сторінок даних на частини буде зменшення навантаження на сторінки контролю.

Вибір того, розбивати масив сторінок даних на кілька підмасивів або використовувати код, що виправляє кілька помилок, залежить від конкретної ситуації та коду.

Для кількісної оцінки надійності коду припускатимемо, що ймовірність збою будь-якої сторінки пам'яті біометричного паспорта в деякий період експлуатації дорівнює p. Тоді захищеність даних можна визначити як

ймовірність Z(p) того, що або жодна сторінка пам'яті не вийде з ладу, або помилка, що відбулася, буде виправлена при декодуванні. Величина Z(p) залежить від використовуваного коду і від способу побудови кодових слів, і є основним показником ефективності коду для моделі організації введення-виведення даних.

2.4.1 Застосування коду Хеммінгу для обробки та зберігання інформації в біометричних паспортах

Раніше було показано, що параметри кодування для кодів Хеммінга в більшості випадків можуть бути підібрані так, щоб кількість сторінок контролю кодового блоку дорівнювала двом. При цьому витрати пам'яті на контрольну інформацію виявляються мінімальними. Тому, оскільки використання кодів Хеммінга в біометричних паспортах передбачає розбиття даних на блоки, сумарна кількість сторінок контролю збільшуватиметься зі збільшенням числа блоків розбиття.

Легко показати, що розбиття блоку кодування на два підблоки збільшує захищеність даних. Для цього позначимо через $Z_1(p)$ захищеність даних коду Хеммінга $C_H(n,2)$. Вона буде обчислюватися за такою формулою:

$$Z_1(p) = (1-p)^n + np(1-p)^{n-1} = (1-p)^{n-1}(1+(n-1)p).$$

При розбитті блоку кодування на два підблоки додаються ще дві сторінки контролю, а сумарна захищеність $Z_2(p)$ буде перебувати як добуток захищеностей кожного з підблоків:

$$Z_2(p) = \left((1-p)^{\frac{n}{2}+1-1}\left(1+\left(\frac{n}{2}+1-1\right)p\right)\right)^2 = (1-q)^n\left(1+\frac{np}{2}\right)^2$$

Отримані величини $Z_1(p)$ та $Z_2(p)$ легко порівнюються. А саме, нерівність $Z_2(p)>Z_1(p)$ рівносильна виконанню нерівності:

$$n(1-q) > 4.$$

Але для кодів Хеммінга мінімальний розмір кодових слів дорівнює трьом, тобто розподіл на підблоки можливий тільки починаючи з n = 4. Отже, отримана нерівність завжди буде виконуватися, коли вона має сенс, тобто коли можливий поділ на блоки.

Таким чином, захищеність даних від помилок буде тим більшою, чим менші розміри кодових блоків [84]. Крім того, зменшення розмірів блоків значно спрощує та скорочує процедури кодування та декодування. Єдиним недоліком такої організації кодування є збільшення витрати пам'яті. Але раніше було показано, що пам'ять біометричних паспортів використовується дуже мало, тому витрата пам'яті при завадостійкому кодуванні перестала бути настільки критичною.

Найбільша захищеність даних досягається при мінімальному розмірі кодових блоків, які будуть складатися з однієї сторінки даних і двох сторінок контролю. Захищеність даних при такому кодуванні становитиме

$$Z(p) = (1 - 3p^2 + 2p^3)^{\frac{N}{3}}$$

де N - загальна кількість сторінок пам'яті.

При цьому обидві сторінки контролю просто дублюватимуть сторінку даних. Отже, процес декодування складається у порівнянні трьох сторінок та виявленні серед них хоча б двох рівних. За такої організації кодування витрату пам'яті буде збільшено втричі.

2.4.2 Застосування коду Ріда-Соломона для обробки та зберігання інформації у біометричних паспортах

Коди Ріда-Соломона можна застосовувати також як і коди Хеммінга, розділяючи пам'ять на блоки, що кодуються, або збільшуючи параметр t. У плані витрати пам'яті обидва методи виявляються рівнозначними. Але збільшення параметра t призводить до більшої захищеності даних, так як даний код буде здатний виправляти помилок t незалежно від їх місця розташування в пам'яті. Єдиним недоліком даного методу є збільшення складності декодування зі збільшенням параметра t.

З іншого боку, метод зменшення розмірів кодованих блоків позбавляє коди Ріда-Соломона переваги перед кодами Хеммінга, які до того ж мають простіший метод декодування.

Отже, застосування кодів Ріда-Соломона є доцільним лише в ситуаціях, коли потрібна максимальна захищеність даних, а метод декодування не вимагатиме великих витрат. Зокрема коди Ріда-Соломона можуть бути використані для даних, що займають невеликий розмір пам'яті.

Висновки по другому розділу

Другий розділ дослідження присвячений методам та алгоритмам обробки, зберігання даних для підвищення надійності системи, в якому було:

- Розроблено модель для випадкового та послідовного вибору сторінок для введення-виводу даних, що дозволяє оцінювати алгоритми завадостійкого кодування інформації в МКПІ з позицій формул для розрахунку ефективності завадостійкого коду, коефіцієнта витрати пам'яті та ступеня захищеності даних.

- Запропоновано методи організації введення-виводу та зберігання даних, що забезпечують надійність SMART-карт, RFID-пристроїв та біометричних паспортів.

- Розглянуто застосування кодів Хеммінга, Ріда-Соломона та локального коду надлишковості в моделі випадкового та послідовного вибору сторінок для введення-виведення даних. Проведено розрахунки складності операцій кодування та декодування даних у персональні ідентифікатори. Вибрано найефективніші модифікації алгоритмів для зберігання персональних даних в ЕСППЗП SMART-карт, RFID-пристроїв, біометричних паспортів та магнітних карт. Наведені таблиці, що показують, що відносна витрата пам'яті кодів Хеммінга зменшується зі збільшенням довжини кодових слів та потужності алфавіту кодування. Наведено графіки, що показують, що ефективність коду прямо пропорційна потужності алфавіту кодування, коди, що мають велику довжину кодових слів, більш ефективно витрачають пам'ять у межах надмірності до 20%.

РОЗДІЛ 3

МОДИФІКАЦІЯ ТА ЗАСТОСУВАННЯ МЕТОДІВ ЗБЕРІГАННЯ ТА ВВЕДЕННЯ-ВИВЕДЕННЯ ДАНИХ В МІКРОКОМП'ЮТЕРНИХ КОМПЛЕКСАХ ПЕРСОНАЛЬНОЇ ІДЕНТИФІКАЦІЇ

3.1. Модифікація методів зберігання та введення-виведення даних у SMART-картках

Розглядається питання модифікацій методів зберігання та введення-виведення даних у SMART-картах, що дозволяє підвищити надійність роботи МКПІ.

Результатами цього розділу є:

- Модифікації файлової системи SMART-карт, та порядок відповідного проектування файлової системи.

- Алгоритми введення-виведення даних, які мають бути реалізовані у SMART-картах з метою забезпечення надійності зберігання інформації.

- Рекомендації по апаратним модифікаціям SMART-карт, що дозволяють підвищити ефективність завадостійкого кодування даних, що зберігаються в них.

Як було показано в другому розділі, забезпечення надійності SMART-карт, може бути досягнуто як застосуванням завадостійких кодів, так і додатковою модифікацією мікроконтролера карти. Іншими словами, всі модифікації в SMART-картах можуть бути поділені на апаратні та програмні.

3.1.1. Модифікації файлової системи SMART-карт

Для забезпечення можливості збереження структури файлової системи запропоновано використання в пам'яті SMART-карток лише систематичних перешкодостійких кодів. При цьому було введено поняття сторінок даних та сторінок контролю. Отже, в пам'яті ЕСППЗП при перешкодах кодування

можуть бути виділені пам'ять даних і контрольна пам'ять, що складається зі сторінок контролю.

Виявлення виходу з ладу сторінки даних чи сторінки контролю передбачає логічну заміну пошкодженої сторінки на нову сторінку пам'яті, що не використовувалася раніше. Таким чином, має бути певний резерв вільних сторінок пам'яті.

Отже, вся пам'ять ЕСППЗП має бути розділена на три логічні розділи:

1. Пам'ять даних - частина пам'яті, що містить сторінки даних, цілісність якої і повинна забезпечуватися в першу чергу.

2. Контрольна пам'ять - частина пам'яті, що містить сторінки контролю.

3. Резервна пам'ять - частина пам'яті, що містить незадіяні в роботі сторінки пам'яті, які можуть бути використані для заміни сторінок даних і контролю, що вийшли з ладу.

Зауважимо, що при подібному розподілі пам'яті код операційної системи SMART-карти або інше програмне забезпечення, що розташовується в пам'яті ЕСППЗП, виявляються розміщеними в пам'яті даних.

Розмір контрольної пам'яті повністю визначається розміром пам'яті даних та використовуваним кодом. Резервна пам'ять, очевидно, представляє собою всю незайняту пам'ять ЕСППЗП. Мінімальний розмір резервної пам'яті, необхідний для забезпечення захисту пам'яті даних від помилок, залежить вже від розмірів пам'яті даних і контрольної пам'яті, але й від особливостей використання пам'яті даних.

Розмір пам'яті даних, отже, і розмір контрольної пам'яті, можуть бути вибрані фіксованими, протягом усього періоду експлуатації, або змінюваними.

У пам'яті даних може бути виділена незмінна частина, яка створюється на етапі ініціалізації чи персоналізації SMART-карти. У цю частину пам'яті входять операційна система та програмне забезпечення у разі їх розміщення у пам'яті ЕСППЗП, а також можуть входити ідентифікаційні та інші дані.

Решту пам'яті даних становлять або змінювані дані, або додається пам'ять, або перезаписуються під час експлуатації.

Таким чином, пам'ять даних можна розділити на три частини:

1. Незмінна частина пам'яті даних - частина пам'яті даних, що записується на етапі ініціалізації або персоналізації SMART-карти.

2. Частина пам'яті даних, що поповнюється - частина пам'яті даних, що містить дані, що додаються в пам'ять в період експлуатації SMART-карти, і більше не змінюються.

3. Частина пам'яті даних, що перезаписується - частина пам'яті даних, що містить дані, які можуть бути перезаписані в період експлуатації SMART-карти.

Кодування зазначених частин пам'яті даних повинно проводитися окремо один від одного. Отже, у контрольній пам'яті також буде виділено три підрозділи, що відповідають підрозділам пам'яті даних. Отримувана логічна структура пам'яті ЕСППЗП зображено на рисунку 3.1.

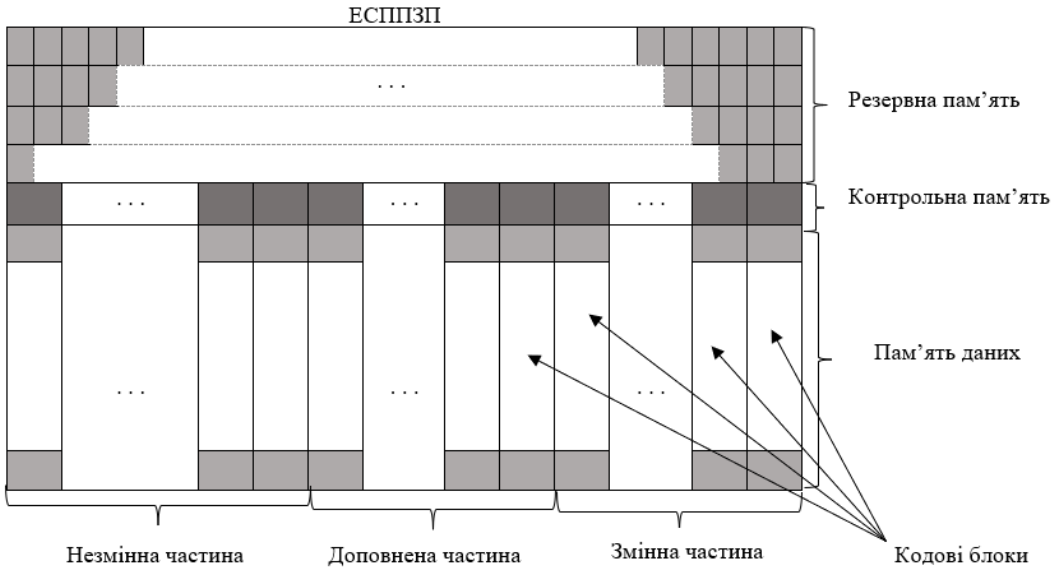

Рис. 3.1. Логічна структура зберігання інформації у SMART-карті

Контрольна інформація, що відповідає незмінній частині даних, визначається також на етапі ініціалізації або персоналізації, і більше не змінюється.

Організація роботи частини пам'яті, що поповнюється, найбільш адекватно описується моделлю послідовного вибору сторінок пам'яті для введення-виведення даних, розібраної на прикладі біометричних паспортів. Тому кодування цієї частини пам'яті даних доцільно виконувати відповідними методами.

Робота частини пам'яті, що перезаписується, найбільш близько описується моделлю випадкового вибору сторінок пам'яті для введення-виведення даних [90, 91]. Кодування цієї частини пам'яті буде виконуватися відповідними методами для зазначеної моделі організації роботи з пам'яттю.

Крім того, вміст пам'яті даних ЕСППЗП може бути розділений за додатковими критеріями:

- Дані можуть бути розділені за групами конфіденційності, коли до однієї групи потрапляють дані, які мають однакові права доступу. Наприклад, ідентифікаційні та автентифікаційні дані швидше за все потраплять у різні групи конфіденційності;
- дані можуть бути розділені та впорядковані за ступенем важливості;
- між даними можуть бути встановлені логічні зв'язки, що доповнюють модель організації роботи з пам'яттю. Наприклад, дані одного файлу у разі його розміщення на кількох сторінках пам'яті означають наявність відповідних логічних зв'язків між сторінками даних.

Кодування даних, що входять до різних груп конфіденційності, в одному інформаційному блоці може послабити захищеність даних від стороннього доступу. Дійсно, раніше було показано, що найбільш відповідні для використання в ЕСППЗП коди мають ті властивості, що будь-яка сторінка контролю кодового блоку обчислюється на основі всіх сторінок даних. Це означає, що будь-яка сторінка контролю потенційно містить у собі у зашифрованому вигляді дані будь-якої сторінки даних з відповідного кодового блоку. Самою собою контрольної інформації, як правило,

недостатньо для розшифрування даних, але знання вмісту контрольної пам'яті може суттєво полегшити зусилля зловмисника. Таким чином, об'єднання даних із різних груп конфіденційності в один кодовий блок потенційно створює загрозу безпеці даних.

Поділ даних за ступенем важливості є важливим для вибору використовуваних методів кодування. Більш важливі дані, очевидно, повинні бути ефективніше захищені від помилок зберігання, можливо, навіть на шкоду захищеності від помилок менш важливих даних.

Логічний зв'язок між даними створюють особливу специфіку організації доступу пам'яті [92]. Певною мірою може бути здійснено розбиття даних на класи, що не перетинаються, в кожен з яких потраплятимуть дані, що записуються або перезаписуються, як правило, разом. З метою скорочення числа циклів стирання/запису найбільш правильним буде кодування пов'язаних в такий спосіб даних в одному кодовому блоці.

Тепер можна сформулювати рекомендації щодо побудови файлової системи пам'яті ЕСППЗП:

1. Пам'ять ЕСППЗП ділиться на три логічні розділи - пам'ять даних, контрольну пам'ять та резервну пам'ять.

Пам'ять даних містить у собі всі дані, що зберігаються у пам'яті ЕСППЗП, крім контрольних даних. Таким чином, цей розділ пам'яті складається зі сторінок даних, і можливо, з невеликої кількості порожніх сторінок пам'яті. Загалом зберігання даних у пам'яті даних може бути організовано так, як це регламентується сучасними стандартами файлових систем SMART-карт.

Розмір пам'яті даних може бути обраним або фіксованим або змінним. Змінним він може бути обраний у тих випадках, коли кількість даних, що розміщуються в пам'яті SMART-карти, значно змінюється протягом періоду експлуатації картки або не може бути точно визначено заздалегідь. Фіксованим розмір пам'яті даних вибирається, якщо обсяг даних, що

зберігається в пам'яті, змінюється незначно - у цьому випадку пам'ять даних у будь-який момент може включати в себе також і порожні сторінки пам'яті.

Контрольна пам'ять повністю складається із сторінок контролю, що відповідають інформаційним блокам даних. Відповідно, розміру контрольної пам'яті визначаються відповідно до кількості інформаційних блоків у пам'яті даних і кодами, що використовуються при їх кодуванні. Кількість інформаційних блоків може бути вибрано фіксованим або збільшується в міру додавання нових даних в пам'ять ЕСППЗП.

Пам'ять ЕСППЗП, що залишилася незайнятою, утворює розділ резервної пам'яті. Сторінки цієї пам'яті або служать для розміщення в них нових даних і контрольної інформації (у разі, коли додавання нових даних призводить до додавання нового інформаційного блоку), або для заміни ними сторінок пам'яті даних, що вийшли з ладу, і контрольної пам'яті.

Резервна пам'ять складається з усіх порожніх сторінок пам'яті (крім даних). Її розмір протягом усього періоду експлуатації, як правило, або залишається незмінним або зменшується. Можлива ситуація, коли розмір резервної пам'яті збільшується за рахунок звільнення сторінок пам'яті даних та контрольної пам'яті.

2. Пам'ять даних поділяється на логічному рівні на інформаційні блоки. Кожен інформаційний блок є набором сторінок пам'яті даних, що включає сторінки даних і можливо порожні сторінки пам'яті. Розподіл пам'яті даних на інформаційні блоки відбувається поетапно при проектуванні файлової системи SMART-карти перед використанням її у конкретній сфері.

По-перше, розділ пам'яті даних ділиться на незмінну, поповнювану і частину, що перезаписується.

Другий поділ йде за групами конфіденційності. Даний етап передбачає досягнення компромісу між вимогами безпеки та наявними можливостями поділу пам'яті на інформаційні блоки (збільшення кількості інформаційних блоків призводить до збільшення розмірів контрольної пам'яті).

Третій поділ даних відбувається за рівнем їх важливості. Бажаним є розміщення даних, що мають різні ступені важливості, у різні інформаційні блоки.

Четвертий етап - облік логічних зв'язків між даними. У випадку зв'язок між даними може призвести до об'єднання різних інформаційних блоків, отриманих на попередніх етапах, в один.

Отримані інформаційні блоки можуть бути розділені, якщо розміри пам'яті дозволяють це.

При додаванні нових даних у пам'ять ЕСППЗП вони можуть бути зараховані або вже до наявних інформаційних блоків, або утворювати нові інформаційні блоки [94]. Таким чином, протягом періоду експлуатації кількість інформаційних блоків може змінюватися як за рахунок видалення даних із пам'яті, так і внаслідок додавання нових даних.

Те, яким чином розподіляти по інформаційних блоках дані, що додаються на запис, також визначається на етапі проектування файлової системи SMART-карти. На рисунку 3.2 схематично представлений етап побудови інформаційних блоків даних під час проектування файлової системи SMART-карт.

Кожен інформаційний блок кодується відповідним кодом, це може бути Q-код Хеммінгу $C_H(N, N-M)$ або код Ріда-Соломона з відповідними значеннями параметрів. Вибір коду для кожного інформаційного блоку залежить від розмірів блоку, необхідної безпеки даних, моделі використання пам'яті, придатної для блоку. Крім того, ці варіанти в сумі суттєво обмежуються розмірами пам'яті ЕСППЗП, обчислювальними ресурсами SMART-картки. Зауважимо, що найбільш простим у реалізації є використання одного коду для кожного інформаційного блоку.

Вирішення завдання вибору параметрів використовуваних кодів залежить від початкових умов завдання таких, як характеристики мікроконтролера SMART-карти, конкретна сфера застосування SMART-карти. Інакше висловлюючись, вибір використовуваних кодів може

відрізнятися для кожного, за конкретного виду SMART-карт та його сфери застосування.

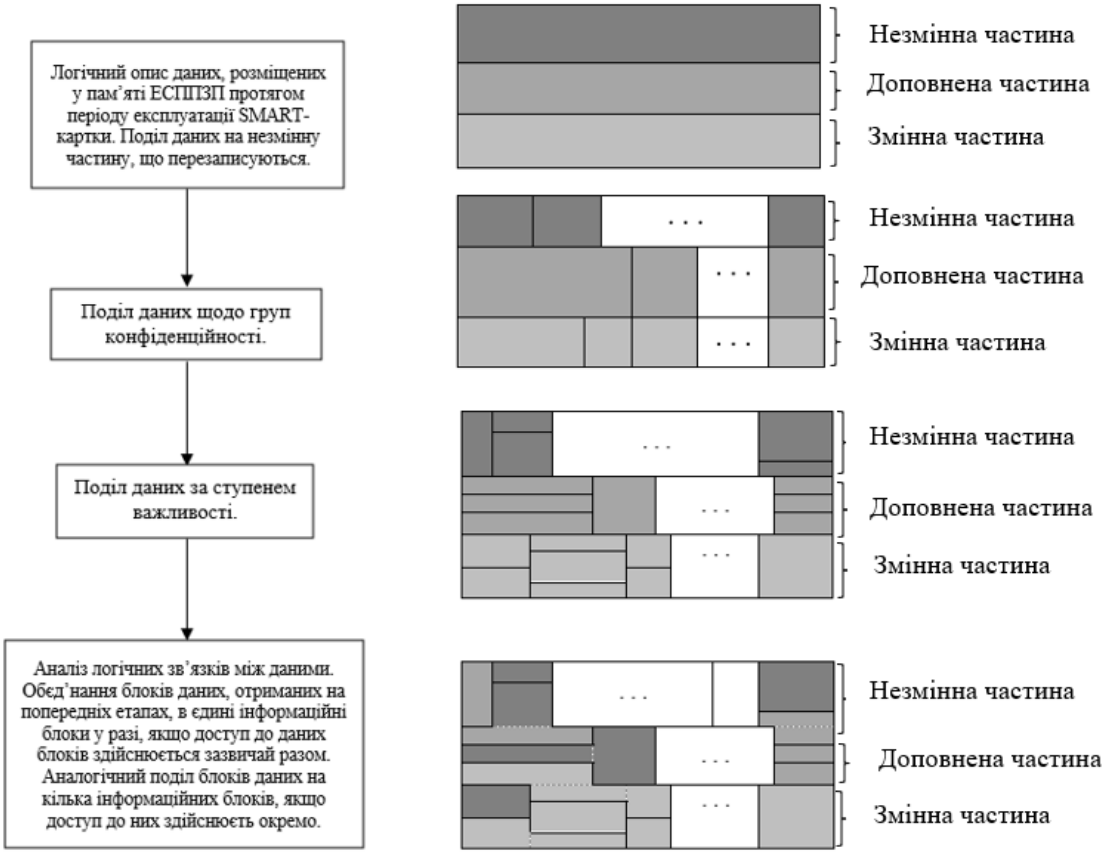

Рис.3.2. Етап поділу даних на інформаційні блоки під час проектування файлової системи SMART-карт

Після виявлення та обчислення помилки на деякій сторінці пам'яті, операційна система SMART-карти (або апаратна схема) повинна спробувати відновлення даних у сторінці пам'яті, тобто, зробити запис відновлених даних. Невдача при спробі перезапису даних говорить про збій сторінки пам'яті та необхідність її заміни. Заміна проводиться на одну із сторінок резервної пам'яті.

У разі мікропроцесорних карт доступ до контрольної пам'яті здійснюється лише операційною системою, і всі процедури кодування та декодування виробляються в межах самої картки. Для карт з мікросхемою пам'яті процедури кодування та декодування можуть здійснюватися або

апаратними схемами, або програмними засобами хост-комп'ютера - у цьому випадку особливу важливість становить максимальний поділ даних у пам'яті даних на інформаційні блоки;

У пам'яті даних має бути також розміщена службова інформація, необхідна для функціонування процедур кодування та декодування [96]. До складу цієї інформації входять параметри кодових блоків такі, як адреси відповідних їм сторінок даних і контролю, параметри коду, що використовуються, якщо передбачається використання в SMART-карті різних кодів.

3.1.2 Реалізація введення-виведення даних у SMART-картах

Відповідно до п. 2.1 можливі дві схеми організації введення-виведення в МКПІ. З міркувань безпеки та економічної ефективності даних найбільш оптимальним є варіант повної реалізації всіх процедур обробки даних у SMART-карті.

Обчислювальні потужності чіпів SMART-карток, як правило, суттєво обмежені, а тому завжди при проектуванні SMART-карток, націлених на застосування в конкретній сфері, має досліджуватися питання про те, наскільки ефективно вони можуть справлятися із завданнями введення-виведення інформаційних блоків даних.

Реалізація процедур кодування та декодування може бути програмною, апаратною та апаратно-програмною.

Перевагою програмної реалізації є порівняльна простота внесення змін до процедури кодування та декодування на етапі розробки. Код реалізованих процедур може бути розміщений як у пам'яті ПЗП, так і в пам'яті ЕСППЗП SMART-картки.

Апаратна реалізація процедур кодування та декодування, навпаки, значно складніше, і буде робити процес проектування SMART-карт менш гнучким, змушуючи використовувати в SMART-картах, що розробляються для різних сфер застосування, більш вузький набір завадостійких кодів. Але

при цьому апаратна реалізація здатна на порядок підвищити швидкість виконання процедур обробки даних.

Найбільш ефективним є апаратно-програмний підхід, коли частина порівняно загальних операцій, властивих процедурам кодування і декодування для перешкод, що розглядаються, кодів, реалізується апаратно, а весь інший функціонал програмно.

Для кодів Хеммінга та Ріда-Соломона загальними, і при цьому найбільш обчислювальними витратними є арифметичні операції в полях Галуа $GF(2')$.

Операція додавання двійкових послідовностей довжини l в полях $GF(2')$ рівносильна виконанню операції $\oplus$ побітового додавання по модулю 2, яка вже реалізована у вигляді команди у більшості процесорів, включаючи мікропроцесори SMART-карт.

Операції ж множення та поділу в $GF(2')$ мають найбільшу обчислювальну складність, і водночас не мають безпосередньої апаратної реалізації у мікроконтролерах SMART-карт.

З проведеного раніше аналізу застосування кодів Хеммінга і Ріда-Соломона в SMART-картах, безпосередньо випливає, що параметр l слід вибирати рівним мірою двійки і не більше 16 (для звичайних розмірів пам'яті ЕСППЗП цього виявляється більш ніж достатньо). Схемні реалізації операцій множення та обігу для полів $GF(2')$ при даних обмеженнях параметра I досить добре досліджені, і можуть бути реалізовані в АЛП мікроконтролера SMART-карти [40], [41], [42].

Зауважимо, що крім апаратної реалізації арифметики кінцевих полів для прискорення виконання цих операцій можуть бути використані також таблиці множення або списки логарифмів елементів поля, що розміщуються в пам'яті ПЗП або ЕСППЗП.

Для SMART-карт з мікросхемою пам'яті процедури кодування та декодування можуть бути реалізовані лише апаратним чином. Крім того,

менш прийнятним з точки зору безпеки та ефективності, але також допустимим є варіант повної або часткової реалізації зазначених процедур на хост-комп'ютері.

Для мікропроцесорних SMART-карт найкращим варіантом є програмна реалізація процедур кодування та декодування з апаратною реалізацією арифметики кінцевих полів $GF(2')$ для відповідних значень I. Простішим, але менш ефективним є варіант програмної реалізації арифметики кінцевих полів $GF(2')$ шляхом застосування таблиць множення та списків логарифмів елементів поля. Найбільшим недоліком такого підходу є фактична неможливість реалізації у пам'яті SMART-карти списку логарифмів елементів поля за $I = 16$.

Як було зазначено раніше, для кодування різних інформаційних блоків у межах однієї SMART-карти можуть бути використані коди Хеммінга та Ріда-Соломона з різними параметрами. Збільшення різноманітності використовуваних завадостійких кодів, у загальному випадку, призводить до ускладнення їхньої спільної реалізації. Причому ускладнюється як реалізація арифметики полів $GF(2')$, якщо використовуються різні значення l, так і реалізація безпосередніх процедур кодування та декодування.

Розглянемо організацію введення-виведення даних у SMART-картах.

До одного і того ж інформаційного блоку даних процедура кодування може бути застосована кілька разів, якщо відбувається запис в нього нових даних або перезапис наявних. В обох випадках як з точки зору збільшення обчислювальної ефективності роботи, так і в плані зменшення числа циклів стирання/запису сторінок контролю, найбільш бажаною є така організація роботи з пам'яттю, щоб процедури запису в окремі сторінки даних максимально об'єднувалися в одну. Іншими словами, процедура кодування має здійснюватися «максимально рідко», але так, щоб при цьому не виникали ситуації невідповідності в кодових блоках між інформаційною та контрольною частинами.

Алгоритм введення даних, ускладнений процедурою завадостійкого кодування, показаний на рисунку 3.3.

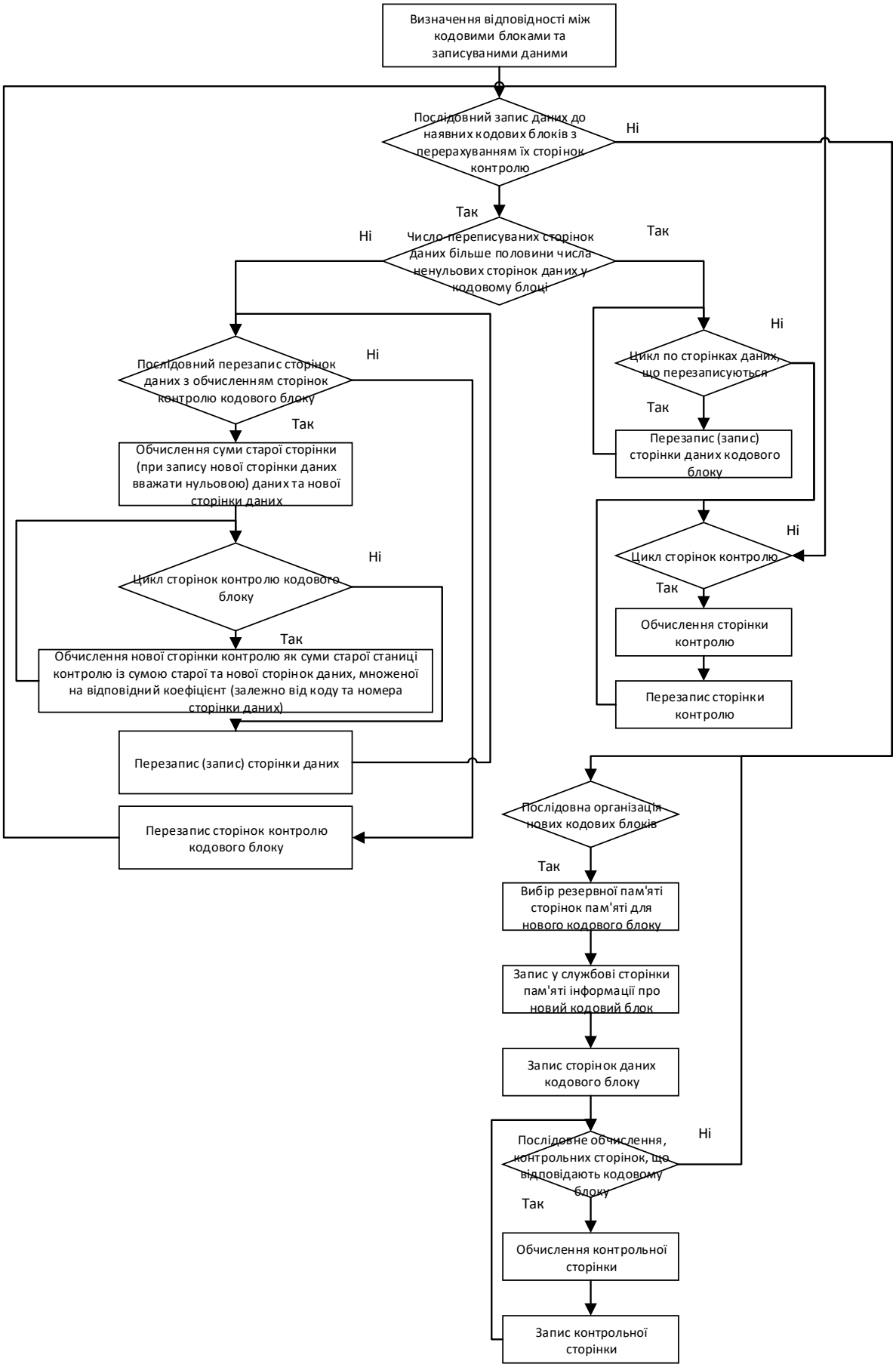

Рис. 3.3. Алгоритм обробки введення даних

Процедура декодування здійснюється після процедури читання даних з пам'яті і здійснюється для всіх кодових блоків, в яких містяться лічені дані. Якщо процедури читання слідують одна за одною, повинні уникати повторного декодування одних і тих же кодових блоків [97]. Для цього процедури читання повинні працювати з різними кодовими блоками, або пропускатися наступні процедури декодування при повторному зверненні до кодового блоку. Крім того, як було показано раніше, процедура декодування повинна здійснюватися у два етапи - перевірка наявності помилки та виправлення помилки. Обов'язковим є лише перший етап декодування, другий виконується лише при виявленні помилки на першому етапі.

Алгоритм виведення даних, ускладнений процедурою декодування, показаний рисунку 3.4.

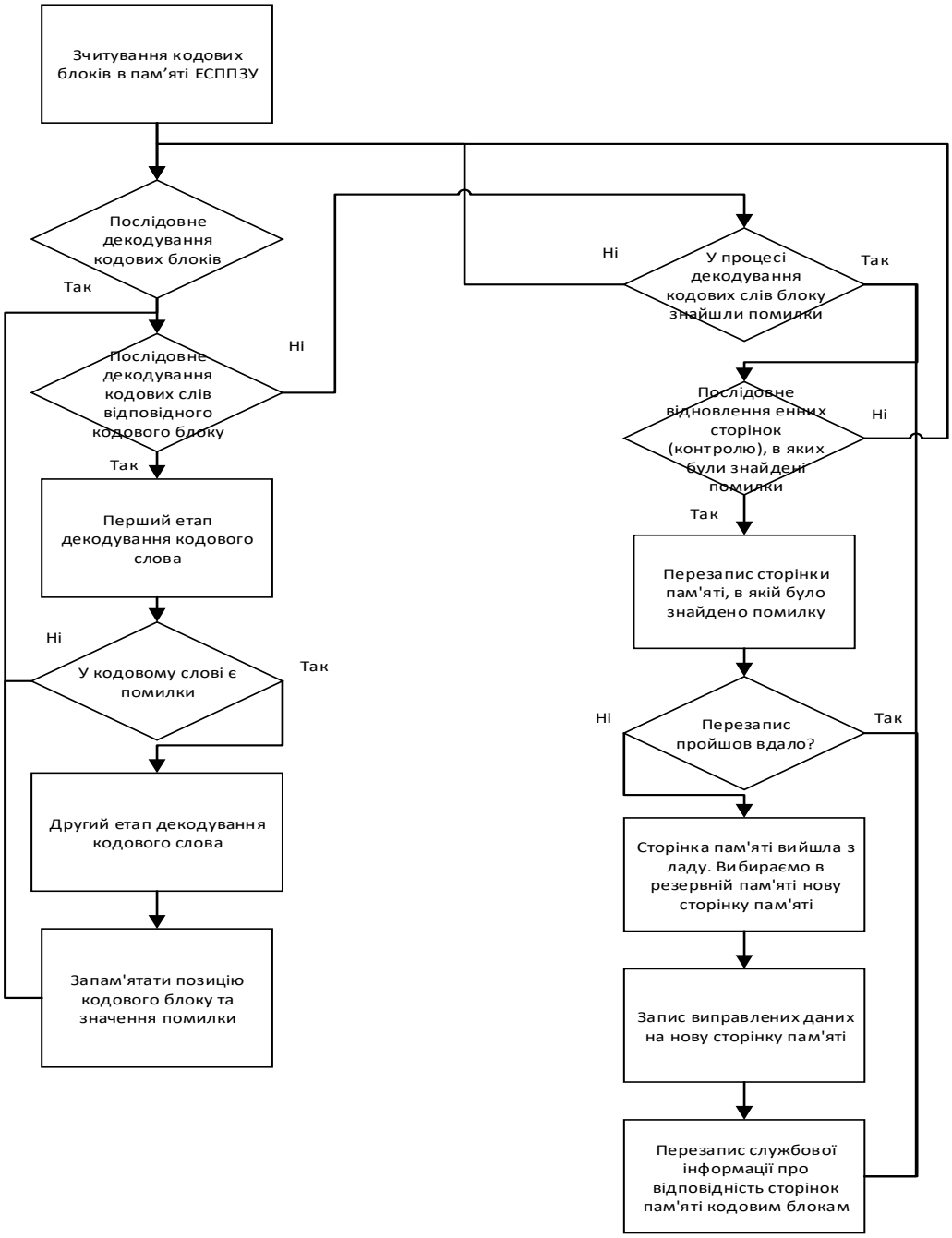

Рис. 3.4 Алгоритм обробки виведення даних

Для кодів Хеммінга перший етап може включати або лише обчислення першого компонента синдрому або обчислення обох компонент синдрому. Перший варіант є більш простим і створює нижчу похибку даних, оскільки факт наявності двох помилок може виявлятися вже не завжди. Але зауважимо, що зазвичай декодування здійснюється відразу для цілого кодового блоку. Перша контрольна сторінка, яку шукаємо, є нічим іншим як побітова сума по модулю 2 всіх сторінок даних інформаційного блоку. У разі збою одразу двох сторінок даних кожна сума, відповідних певному біту,

залишатиметься незмінною з ймовірністю $^1/_2$. Імовірність того, що виявиться незмінною вся сторінка контролю, дорівнює $^1/_{2^{8L}}$, де L - розмір сторінки пам'яті в байтах. Таким чином, ймовірність не виявлення факту двох помилок є вкрай малою, а тому в перший етап декодування для кодів Хеммінга досить включати тільки обчислення першої сторінки контролю.

Для кодів Ріда-Соломона перший етап декодування завжди полягає у обчисленні компонентів синдрому.

Значне спрощення алгоритмів введення-виведення даних, а також зменшення розміру контрольної пам'яті може бути досягнуто, якщо в мікроконтролері ЕСППП будуть реалізовані інші програмно-апаратні можливості виявлення збою сторінок пам'яті. У такій ситуації для подальшого обчислення одноразової помилки достатньо знати суму всіх сторінок даних кодового блоку. Іншими словами, достатньо використання локального коду надлишковості коду.

Резюмуючи, позначимо пропоновані апаратно-програмні модифікації SMART-карт, що забезпечують ефективне виконання процедур введення-виведення:

- Апаратна реалізація арифметики кінцевих полів $GF(2')$ для l = 2,4,8,16 або для деяких із зазначених значень l. Дана модифікація мікроконтролерів SMART-карт дасть найбільший ефект прискорення роботи процедур кодування та декодування.

- Альтернативою апаратної реалізації арифметики полів $GF(2')$ є створення в пам'яті ПЗП та ЕСППЗП таблиць множення або списків логарифмів елементів поля.

- Програмно-апаратна реалізація функції виявлення збоїв у сторінках пам'яті ЕСППЗП. Дана модифікація дозволить використовувати як перешкодостійкий локальний код надлишковості, що дозволить максимально спростити процедури введення-виведення, а також суттєво скоротити розмір контрольної пам'яті.

3.2 Модифікація методів зберігання даних у магнітних картах

Результатом розділу є три альтернативні модифікації зберігання даних, що дозволяють підвищити надійність зберігання даних без істотних змін в організації роботи карт.

Раніше було показано, що підвищення перешкодозахищеності даних, що зберігаються в магнітних картах, може бути досягнуто шляхом розширення використовуваного в них завадостійкого коду. Зазначене перетворення коду передбачає додавання даних додаткових контрольних символів.

Найбільш логічним шляхом модифікації є додавання нових контрольних символів в кінці магнітних доріжок після символів LRC. Дана модифікація передбачає внесення змін до стандартів ISO-7811, що регламентують структуру даних, що розміщуються на магнітних доріжках карток.

Зауважимо, що деякі символи є зарезервованими, і їх використання в блоці даних заборонено, в той час як значення контрольних символів можуть бути довільними. Символ LRC розміщується після символу закінчення блоку даних, тому для нього допустимі будь-які значення. Таким чином, єдиним місцем розташування додаткових контрольних символів є розміщення їх разом з символом LRC.

Розглянуто три модифікації коду Хеммінга, які можна використовувати в магнітних доріжках. Отже, можна сформулювати три альтернативні пропозиції щодо модифікації зберігання даних у пам'яті магнітних карток, з метою підвищення їх надійності:

1. Наприкінці кожної магнітної доріжки після символу LRC розміщуються два додаткові контрольні символи.

Для першої магнітної доріжки значення цих символів обчислюються за допомогою укороченого коду Хеммінга $C_H(79,76)$ з кодуванням алфавітом $GF(64)$. Цей код виходить із систематичного коду $C_H(4161,4158)$

видаленням необхідної кількості символів наприкінці кодових слів. Як кодовані слова розглядаються послідовності символів магнітної доріжки без бітів парності. Додаткові контрольні символи доріжки знаходяться як другий і третій контрольні символи кодового слова з доданими до них старшими парами парності.

Для другої магнітної доріжки додаткові контрольні символи обчислюються аналогічно за допомогою коду Хеммінга $C_H(40,37)$ з алфавітом кодування $GF(16)$.

Додаткові контрольні символи третьої доріжки таким же чином виходять за допомогою коду Хеммінга $C_H(107,104)$ з кодуванням алфавітом $GF(16)$.

2. Наприкінці першої магнітної доріжки після символу LRC розміщуються три додаткові контрольні символи.

Для обчислення контрольних символів кожен символ даних на магнітній доріжці ділиться на три 2-бітові частини і один біт парності. З отриманих частин всіх символів складаються три інформаційні слова над полем $GF(4)$. Кожне слово кодується укороченим кодом Хеммінгу $C_H(79,75)$ з алфавітом кодування $GF(4)$. Цей код виходить відповідним чином з коду Хеммінга $C_H(85,81)$.

Складаючи у 6-бітові слова відповідні контрольні символи трьох кодових слів, отримуємо чотири 6-бітові символи. Додаючи до них біти парності, отримуємо символ LRC та три додаткові контрольні символи, які і розміщуються разом із символом LRC.

В кінці другої магнітної доріжки після символу LRC також розміщуються три додаткові контрольні символи. Для їх обчислення кожен символ даних на магнітній доріжці поділяється на дві 2-бітові частини та один біт парності. З отриманих частин всіх символів складаються два інформаційних слова над полем $GF(4)$, які кодуються далі укороченим кодом Хеммінга $C_H(40,36)$ з алфавітом кодування $GF(4)$. Цей код виходить із коду

Хеммінга $C_H(85,81)$ видаленням необхідної кількості символів наприкінці кодових слів.

Складаючи у 4-бітне слово відповідні контрольні символи двох кодових слів, отримуємо чотири 4-бітні символи [99]. Додаючи до них біти парності, отримуємо символ LRC і три додаткові контрольні символи, що розміщуються після символу LRC.

Наприкінці третьої магнітної доріжки додаються чотири додаткові контрольні символи. Їхня побудова робиться також як і для другої доріжки, за винятком того, що використовується укорочений код Хеммінга $C_H(107,102)$, що виходить із систематичного коду Хеммінга $C_H(341,336)$.

3. Наприкінці першої та третьої доріжок розміщується 6 додаткових контрольних символів, наприкінці другої - 5 додаткових контрольних символів. Їх обчислення здійснюється так як і для другого варіанта модифікацій, з тією різницею, що як алфавіт кодування використовується поле $CF(2)$, укорочені коди Хеммінга для першої та третьої доріжок виходять з коду $C_H(127,120)$, а укорочений код для другої доріжки будується із коду Хеммінга $C_H(63,57)$.

Запропоновані методи зберігання даних у магнітних картах, мають наступні переваги перед використовуваним в даний час:

- Запропоновані методи кодування мають більшу ефективність виявлення помилок, дозволяючи гарантовано відстежувати некоректність даних при більш значному числі помилок у них. В даному сенсі, ефективності пропонованих модифікацій зростають з першого варіанту до третього. Зі збільшенням ефективності також збільшується необхідна кількість додаткових контрольних символів. Але в цілому воно залишається невеликим і не перевищує 15%.

- Запропоновані модифікації не передбачають сильних змін у структурі даних, що розміщуються в пам'яті картки, будучи лише невеликим доповненням до існуючих стандартів захисту даних, що зберігаються в пам'яті магнітних карток, від помилок.

3.3 Модифікація методів зберігання та введення-виведення даних у мікрокомп'ютерні комплекси персональної ідентифікації на основі технології RFID

У цьому розділі проводиться аналіз модифікацій, що дозволяють знизити енерговитрати на процедури зберігання та введення-виведення даних, що реалізуються в МКПІ на основі технології RFID.

Результатом даного розділу є схематичні описи процедур введення-виведення в МКПІ при їх повному або частковому розміщенні на хост-комп'ютері.

Найбільшими можливостями в плані застосування методів завадостійкого кодування до даних мають RFID-мітки з електронним носієм пам'яті. Однобітові мітки та SAW-мітки значно менш гнучкі носії інформації - більш висока захищеність даних у них може бути досягнута тільки шляхом значних апаратних модифікацій. Розглянемо лише RFID-мітки з електронним носієм пам'яті.

Даний тип МКПІ, як було показано раніше, дуже схожий на SMART-карти, а тому і методи зберігання та введення-виводу для них переважно збігаються.

Також як і для SMART-карт, пам'ять ЕСППЗП RFID-систем розділяється спочатку на пам'ять даних, контрольну пам'ять і резервну пам'ять, а потім ділиться на інформаційні блоки.

Головна відмінність від SMART-карт полягає у відсутності безперебійного джерела живлення, що тягне за собою необхідність додаткового критерію ефективності завадостійких кодів, малої енерговитратності процедур зберігання та введення-виведення [101]. Було запропоновано три способи, якими можна досягти виконання цієї умови: зменшення розмірів інформаційних блоків, перенесення процедур кодування на персональний ідентифікатор, апаратна реалізація процедур кодування. Всі ці способи мають певні недоліки, і вибір серед них може бути здійснений

тільки при аналізі технічних особливостей RFID-системи, сфери її застосування та вимог. Даний аналіз може бути зроблений за такими критеріями:

- Аналіз файлової системи RFID-пристрою, даних, що розміщуються в ньому як перед початком експлуатації, так і в період виявлення можливостей зменшення розмірів інформаційних блоків для даних, що зберігаються в пам'яті ЕСППЗП RFID-пристрою.

- Аналіз вимог безпеки, що висуваються до системи RFID-ідентифікації. З'ясування питання, чи можуть бути повністю або частково перенесені процедури кодування та декодування на хост-комп'ютер.

- Аналіз можливостей апаратної реалізації найбільш трудомістких обчислень у процедурах кодування та декодування (як і для SMART-карт такими є арифметичні операції в полях Галуа $CF(q)$).

На основі даного аналізу може бути знайдено оптимальне рішення того, як і які засоби підвищення ефективності використовувати в системі RFID-ідентифікації.

Алгоритм введення-виведення даних для МКПІ RFID у разі розміщення алгоритмів кодування декодування інформації в персональному ідентифікаторі мають такі ж алгоритми, як на рисунках 3.3 та 3.4.

Метод введення даних може бути реалізований з розміщенням алгоритмів кодування/декодування інформації на хост-комп'ютері відповідно до схеми, представленої на рисунку 3.5.

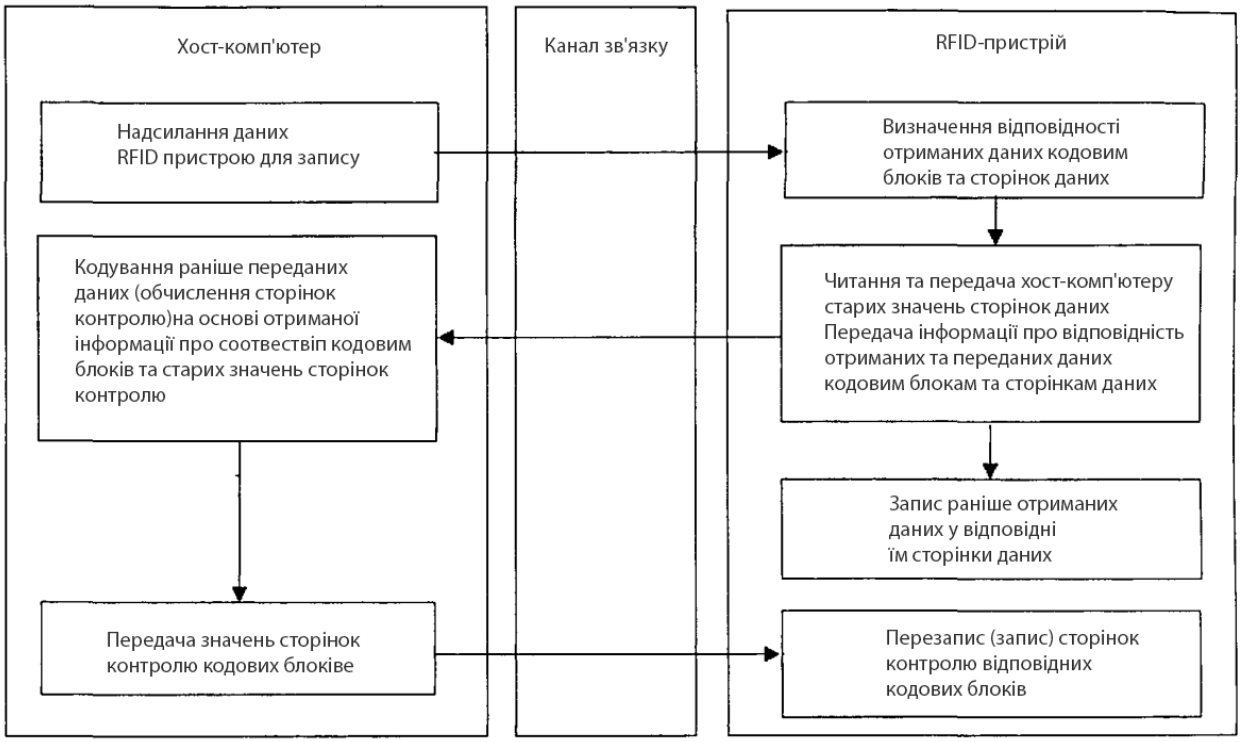

Рис. 3.5. Організація введення інформації під час розміщення алгоритмів
забезпечення надійності на хост-комп'ютері

Метод виведення даних із використанням алгоритмів декодування інформації бути повністю або частково розміщено на хост-комп'ютері. Організація роботи процедури при повному її розміщення на хост-комп'ютері зображено на рисунку 3.6.

Процедура виведення даних з використанням алгоритмів декодування інформації, що частково розміщена на хост-комп'ютері, схематично зображена на рисунку 3.7.

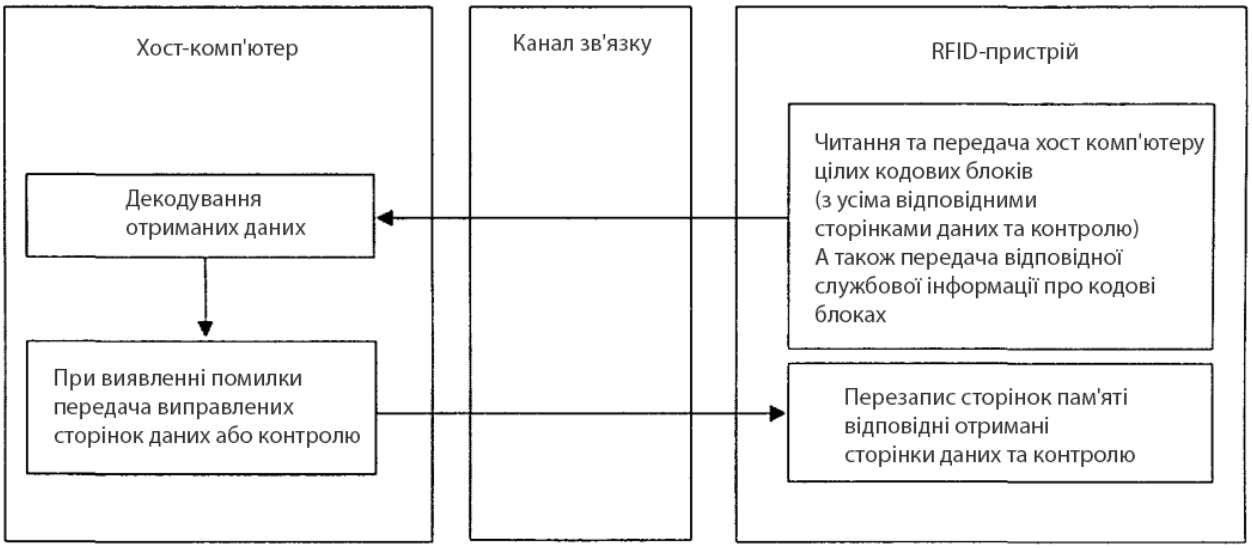

Рис.3.6. Організація виведення інформації при повному розміщенні алгоритмів забезпечення надійності на хост-комп'ютері

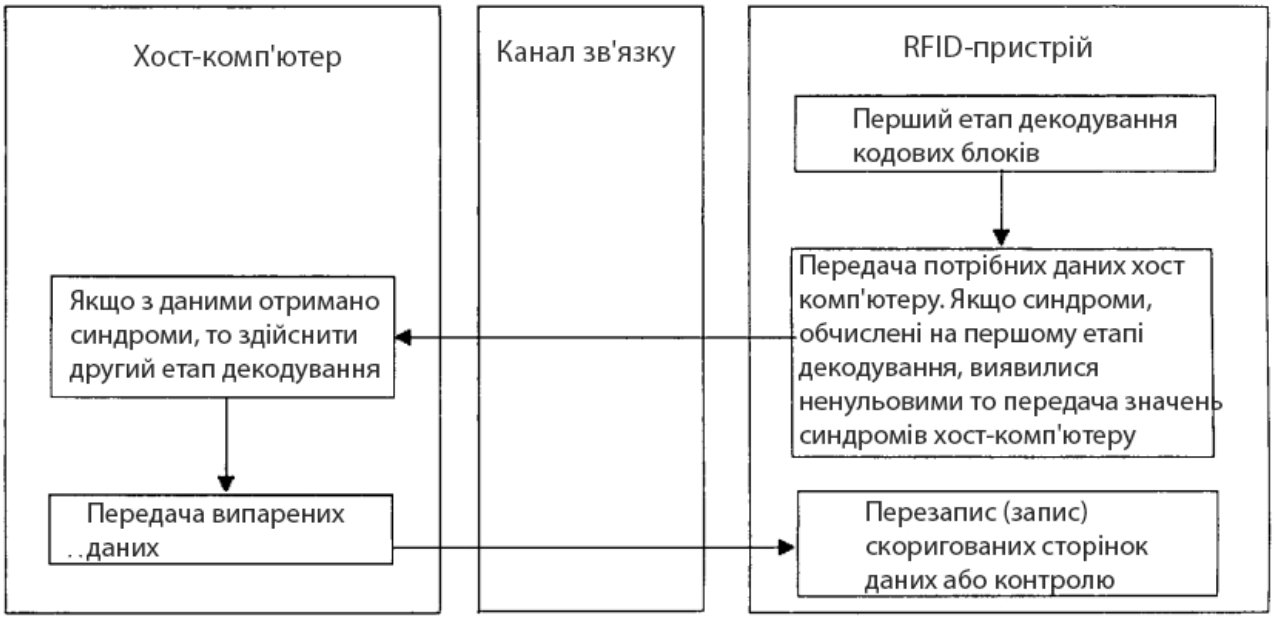

Рис.3.7. Організація виведення інформації при частковому розміщенні алгоритмів забезпечення надійності на хост-комп'ютері

3.4 Модифікація методів зберігання та введення-виведення даних у біометричних паспортах

Результатами цього розділу є:

- Модифікація файлової системи біометричних паспортів з метою підвищення захищеності даних, що зберігаються у пам'яті паспорта, від помилок;

- Опис процедур введення-виведення даних у біометричних паспортах.

Як було зазначено раніше, біометричні паспорти можуть розглядатися як безконтактні SMART-картки. До них застосовні багато з розглянутих вище модифікацій, що підвищують стійкість до перешкод SMART-карт і RFID-пристроїв. З іншого боку, також було показано, що методи декодування, які застосовуються в біометричних паспортах, можуть мати специфіку.

Вище було показано, що проектування файлової системи SMART-карт має включати етап розбиття даних на інформаційні блоки. Проведемо даний етап дослідження даних біометричного паспорта:

1. У пам'яті ЕСППЗП біометричного паспорта зберігаються ідентифікаційні дані власника та списки доступу. По-перше, це загальні ідентифікаційні дані, записані на паспорті. Вони сформовані у пам'яті ЕСППЗП як окремий файл. По-друге, ідентифікаційні дані включають біометричні дані.

Список доступу складається з віз, кількість яких може змінюватися протягом періоду експлуатації паспорта.

2. Всі ідентифікаційні дані є незмінними, і, крім того, передбачається можливість додавання в пам'ять нових біометричних даних [103]. Таким чином, у пам'яті даних можуть бути виділені незмінна пам'ять, поповнювана і пам'ять, що перезаписується, відповідно візам.

Розмір незмінної пам'яті разом із кодом програмного забезпечення становитиме трохи більше 3-4 Кбайт. Розмір поповнюваної пам'яті, очевидно, варто вибрати змінним і спочатку рівним 0. Теж саме вірно і для розділу пам'яті даних, що перезаписується. Зауважимо, що розділ, що перезаписується, може бути найбільшим, і мати розмір навіть більше 20 Кбайт.

3. Дані, що розміщуються у біометричних паспортах, мають три рівні конфіденційності. Першу групу конфіденційності становлять загальні ідентифікаційні дані. Другий рівень конфіденційності мають візи, але кожна віза може бути зашифрована своїм особливим чином. Тому кожна віза становитиме свою групу конфіденційності. Останню групу конфіденційності складають біометричні дані.

4. Найбільш важливими даними, які зберігаються в біометричному паспорті, є ідентифікаційні дані. За ними слідують списки доступу.

5. Отримано розбиття даних на такі інформаційні блоки: загальні ідентифікаційні дані, кожна віза, набір біометричних даних. Крім біометричних даних, всі дані були розбиті на окремі логічно незалежні файли.

Блок біометричних даних може бути розбитий на окремі файли, або залишений у вигляді єдиного інформаційного блоку. Біометричні дані становлять поповнюваний розділ пам'яті, тому для них можливими виявляються збої відразу кількох сторінок пам'яті. Раніше було показано, що в такій ситуації найбільш доцільним є або розбиття даних на дрібніші інформаційні блоки, або використання коду, що виправляє кілька помилок. З іншого боку, біометричний паспорт є пристроєм, і тому велику важливість набувають рівні енерговитрат, до яких наводять процедури кодування та декодування. Тому зупинимо свій вибір на розбитті єдиного інформаційного блоку біометричних даних, на блоки, відповідні кожному файлу біометричних даних.

Зауважимо, що в українських біометричних паспортах зараз міститься лише один файл з біометричними даними, а тому для нього обидва вибори поки що рівнозначні.

Отримане розбиття даних біометричного паспорта на інформаційні блоки зображено на рисунку 3.8.

Загальні ідентифікаційні дані (ім'я. пол. країна тощо)	Біометричні дані 1	Біометричні дані 2	…	Біометричні дані m

Віза 1	Віза 2	…	Віза m

Рис. 3.8. Розбиття на інформаційні блоки даних біометричного паспорта

Роботу незмінного і поповнюваного розділів пам'яті, що містять ідентифікаційні дані, найбільш адекватно описує модель послідовного вибору сторінок пам'яті для запису. Раніше за допомогою даної моделі у випадку біометричних паспортів було показано, що найбільш ефективним буде використання кодів Хеммінга для інформаційних блоків з однієї сторінки даних, або використання кодів Ріда-Соломона, які виправляють кілька помилок [101]. Оскільки біометричний паспорт є RFID-пристроєм, зупинимо свій вибір на менш енерговитратних кодах Хеммінга. Отримуємо, що інформаційні блоки, що відповідають ідентифікаційним даним, кожен складатиметься з однієї сторінки даних.

Список доступу міститься в розділі пам'яті, що перезаписується, робота якого найкраще описується моделлю випадкового вибору сторінок для запису. Розбиття на менші інформаційні блоки в цьому випадку не таке доцільне - більш прийнятним способом підвищення захищеності даних є використання кодів Ріда-Соломона, що виправляють не менше двох помилок. Приймемо як код, що використовується для кодування віз, код Ріда-Соломона з параметром $t \geq 2$. Розміри файлів віз не дуже великі, а тому використання для них кодів Ріда-Соломона не повинно призвести до надмірних енергетичних витрат, у той час як ймовірність помилок зросте.

Отже, ідентифікаційні дані біометричного паспорта кодуватимуться простим подвійним дублюванням даних, а візи кодом Ріда-Соломона, що виправляє більше однієї помилки. На рисунку 3.9 зображено відповідну логічну структуру пам'яті ЕССПЗП біометричного паспорта.

ЕСППЗП

Загальні ідентифікаційні дані (ім'я. пол. країна і т.д.)	Біометричні дані 1	Біометричні дані2	…	Біометричні дані m	Ідентифікаційні дані
Загальні ідентифікаційні дані (ім'я стать країна і т д) Копія 1	Біометричні дані1 Копія 1	Біометричні дані 2 Копія 1	…	Біометричні дані m Копія 1	Копії ідентифікаційних даних
Загальні ідентифікаційні дані (ім'я поп країна і т.д.) Копія 2	Біометричні дані1 Копія 2	Біометричні дані2 Копія 2	…	Біометричні дані m Копія 2	Список доступу
Віза 1	Віза 2	…		Віза m	Контрольні частини файлів-віз
		…			Резервна пам'ять

Рис.3.9. Логічна структура зберігання інформації у біометричному паспорті

Процедура введення біометричних даних є простим дублюванням даних. Але при цьому додавання нових біометричних даних не є звичайною операцією в ході експлуатації біометричного паспорта. Тому процедуру кодування можна розглядати лише для віз. Загалом алгоритм введення даних здійснюватиметься відповідно до алгоритму рисунку 3.3.

Процедура виведення для ідентифікаційних даних є порівнянням трьох сторінок пам'яті, і може бути описана алгоритмом як на рис. 3.10:

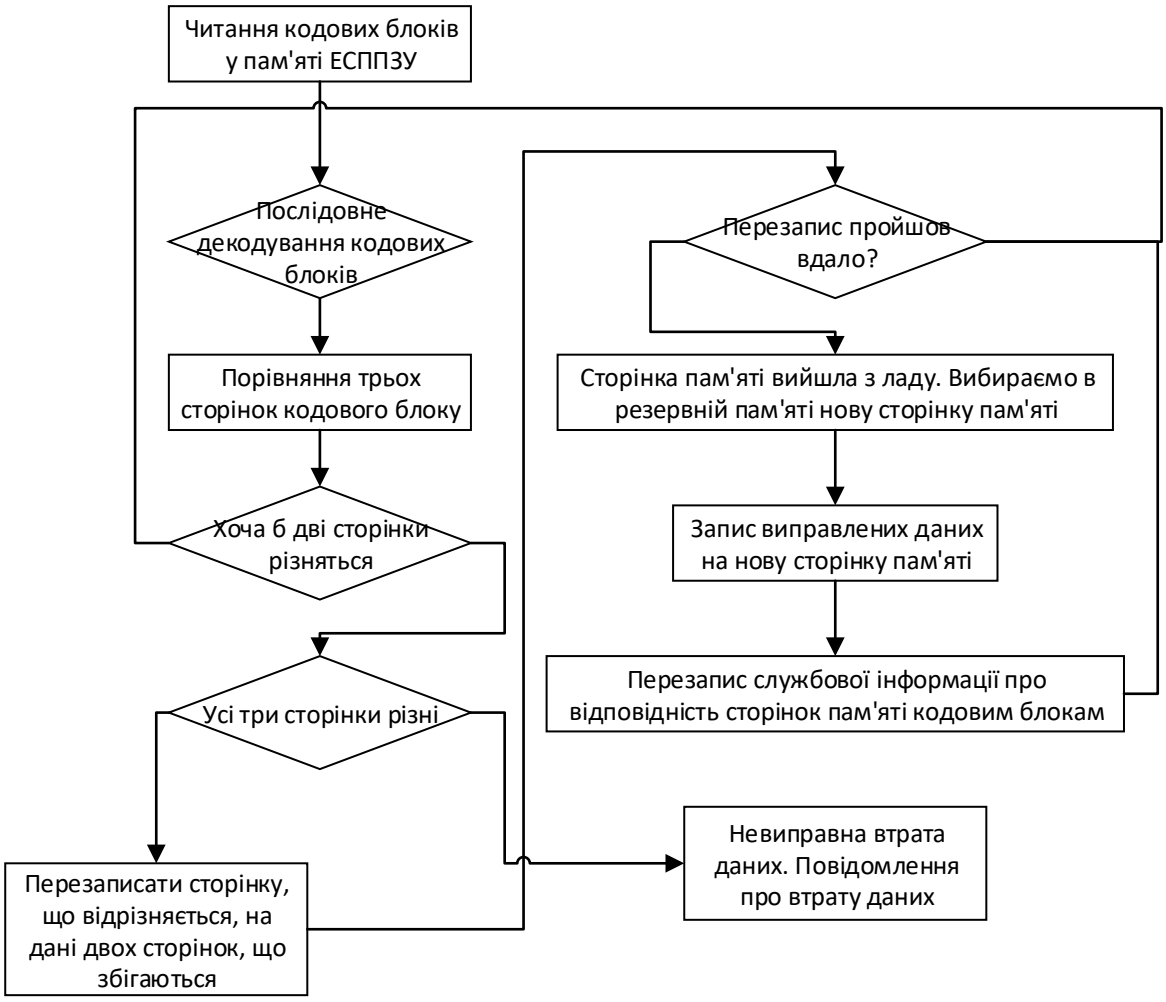

Рис. 3.10. Алгоритм виведення ідентифікаційних даних біометричного паспорта

3.4.1 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації на базі SMART-карт

Алгоритми та методи обробки, зберігання та введення-виведення інформації можуть бути успішно реалізовані з метою підвищення SMART-карток, що використовуються в різних сферах діяльності. У цьому розділі покажемо дві з таких можливих реалізацій для так званих міських соціальних карток та для SIM-карток апаратів стільникового зв'язку.

Міські соціальні карти

Міські соціальні карти є багатофункціональними ідентифікаційними картками, які на сьогоднішній день є частиною інфраструктури, що активно

розвивається. Міські карти корисні всій міській спільноті, в рамках якої вони діють. Місцевій владі вони дозволяють більш точно та ефективно розподіляти та надавати соціальні послуги та пільги. Місцевим мешканцям вони дозволяють замінити кілька специфічних SMART-карток однією багатофункціональною.

У загальному випадку міська карта містить у собі ідентифікаційні та особисті дані власника, і може виступати як ідентифікаційне посвідчення, банківська картка, платіжна картка, страхова, пільгова картка, картки лояльності та ін. [104]. Крім того, міська карта може містити в собі різну додаткову корисну інформацію: карту міста, маршрути руху громадського транспорту, адреси медичних установ.

Таким чином, дані, що зберігаються в пам'яті міської карти, умовно можна розділити на ідентифікаційні та особисті та додаткові загальнодоступні соціальні дані.

Втрата або спотворення ідентифікаційних та особистих даних, що зберігаються в пам'яті міської карти, веде до необхідності їх відновлення шляхом звернення до відповідних сервісних центрів або навіть повної заміни картки. Обидва варіанти розвитку подій створюють значні незручності як для користувача міської картки, так і для сервісних центрів, що обслуговують картку.

Помилки, що виникають у додаткових даних карти, не настільки критичні, але все ж таки можуть створювати за певних умов значні незручності для користувача.

Виходом з цієї ситуації може бути застосування методів завадостійкого кодування в картах, що дозволить виправляти помилки, що виникають, а також сигналізувати власнику карти про появу помилок у пам'яті карти, як виправлених, так і некоректованих.

Наростаюча статистика появи помилок у карті може свідчити про наближення закінчення терміну служби картки та необхідність її своєчасної заміни. При цьому повідомлення про статистику виправлених або

невиправлених помилок у карті повинні передаватися хост-комп'ютеру, а від нього сервісному центру, який вже прийматиме рішення про заміну картки.

Таким чином, застосування методів та алгоритмів, розроблених у другому розділі, в міських картах дозволить не тільки підвищити захищеність даних від помилок, а й у багатьох випадках проводити своєчасну заміну карток до їхнього остаточного виходу з ладу.

Зауважимо, що з метою безпеки в повідомленнях хост-комп'ютеру повинна передаватися лише інформація, яка не містить відомостей про особисті та ідентифікаційні дані користувача [105]. Наприклад, інформація про кількість сторінок пам'яті соціальної SMART-карти, що вийшли з ладу, інформація про наявність непоправних втрат ідентифікаційних даних.

Реалізація системи захисту від помилок у міських соціальних картах схематично представлена на рисунку 3.11.

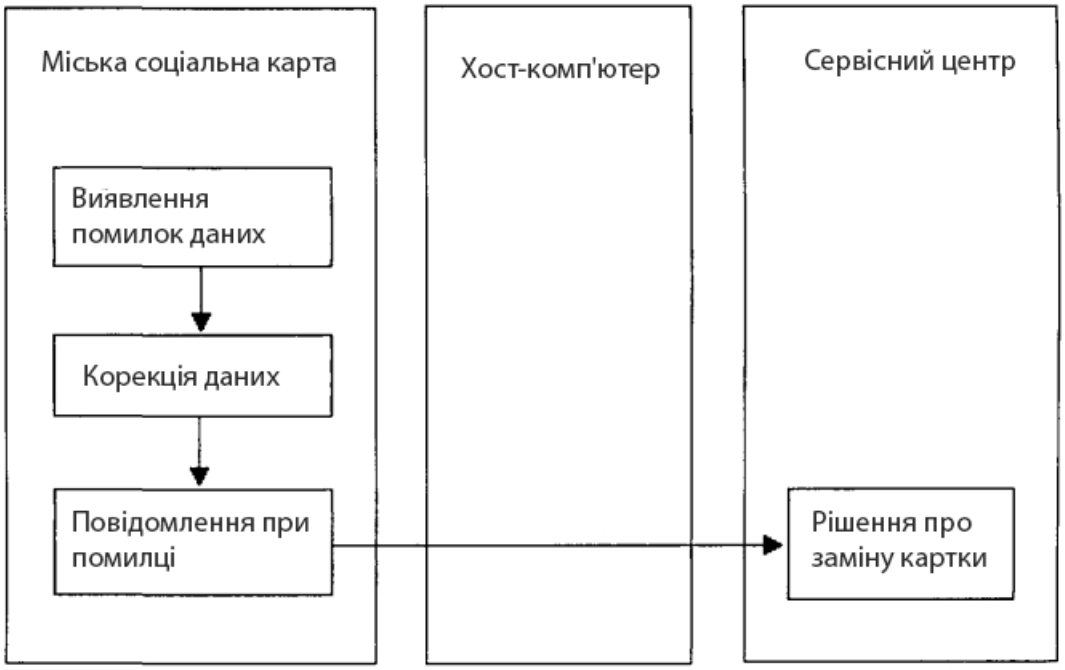

Рис. 3.11. Система захисту від помилок у міських соціальних картах

Перелічимо основні переваги, які надасть використання методів завадостійкого кодування у міських соціальних картах:

- Підвищення рівня захищеності даних, що зберігаються у пам'яті карти, від помилок. Виправлення помилок, що виникають в результаті виходу з ладу сторінок пам'яті картки або з інших причин.

- Вищий рівень контролю цілісності даних, збережених у пам'яті соціальної карти. Можливість виявлення спотворень даних до виникнення колізій під час роботи з ними.
- Можливість моніторингу ресурсів працездатності кожної міської картки: виявлення карток, що ще не вийшли з ладу, але наближення до закінчення терміну служби, заміна ненадійних карт на нові.
- Можливість збільшення гарантійного терміну служби карт за рахунок підвищення захищеності даних від помилок та моніторингу їхньої працездатності в пізні періоди їх експлуатації.

Можна назвати такі недоліки використання методів завадостійкого кодування у міських соціальних картах:

- додаткові витрати пам'яті ЕСППЗП карт, отже, більш високі вимоги до розміру пам'яті карт;
- незначне підвищення собівартості використовуваних SMART-карток;
- Можливе зменшення продуктивності карток при роботі з великими блоками даних (карти міста, маршрути руху транспорту).

Остаточно, можна зробити висновки, що застосування методів завадостійкого кодування дозволить значно підвищити надійність системи міських соціальних карт, надаючи також значні можливості для розвитку даної інфраструктури.

SIM-карти апаратів стільникового зв'язку

Найбільш поширеним застосуванням SMART-карт служить використання їх як SIM-карт апаратів стільникового зв'язку.

SIM-карта є невід'ємним елементом мережі стільникового зв'язку, дозволяючи мобільному телефону проводити реєстрацію в мережі оператора, користуватися послугами передачі даних та голосу, підтверджувати особу абонента. SIM-карти мають, крім модулів пам'яті, повноцінний мікропроцесор з арифметико-логічним пристроєм, пристроєм управління, регістрами, криптографічним співпроцесором. Передбачають наявність

спеціалізованої файлової системи GSMFS (Global System for Mobile Communications File System). Файлова система глобальної системи мобільного зв'язку, що забезпечує великий набір сервісних операцій та засобів безпеки та передбачає розмежування доступу до інформації.

Найбільш ефективна модель організації роботи пам'яті цього виду карт є модель випадкового вибору сторінок пам'яті для запису, коли сторінки пам'яті багаторазово перезаписуються з тією чи іншою ймовірністю. Кожна подія періодичної або нормальної реєстрації на базових станціях, SMS-повідомленнях чи дзвінках, породжує зміну або додавання інформації до пам'яті SIM-карти. Фактично це означає, що з великою ймовірністю, в пам'яті ЕСППЗП SIM-карти є сторінки пам'яті, що перезаписуються при кожній події.

Таким чином, пам'ять ЕСППЗП SIM-карт містить високо навантажені сторінки пам'яті, і найбільш імовірною помилкою є їхній вихід з ладу. З іншого боку, всі дані SIM-карти, необхідні для функціонування мережі оператора зв'язку, містяться тільки в ній, і як правило, ніде не продубльовані. Тому, забезпечення безпеки даних карти, і виявлення необхідності її заміни до непоправної зміни цілісності даних, є досить важливим завданням.

Забезпечення цілісності даних може бути досягнуто шляхом застосування методів завадостійкого кодування моделі випадкового вибору сторінок для запису [98]. Дані методи дозволять з великою ймовірністю виявити та виправити перший збій даних у карті.

Так як дані, що зберігаються в пам'яті карти, ніде не продубльовані, вони не можуть бути відновлені у разі появи досить великої кількості помилок. Тому необхідним сценарієм у цьому випадку є виявлення помилки в SIM-карті, корекція даних, оповіщення оператора зв'язку за протоколом мобільної телефонії OTASP про збій у SIM-модулі та оповіщення користувача до виникнення ситуації безповоротної втрати даних. Отже, при виявленні першого збою даних абонент повинен отримати повідомлення про необхідність заміни SIM-карти на нову.

Реалізація системи захисту від помилок у SIM-картах апаратів стільникового зв'язку схематично представлена рисунку 3.12.

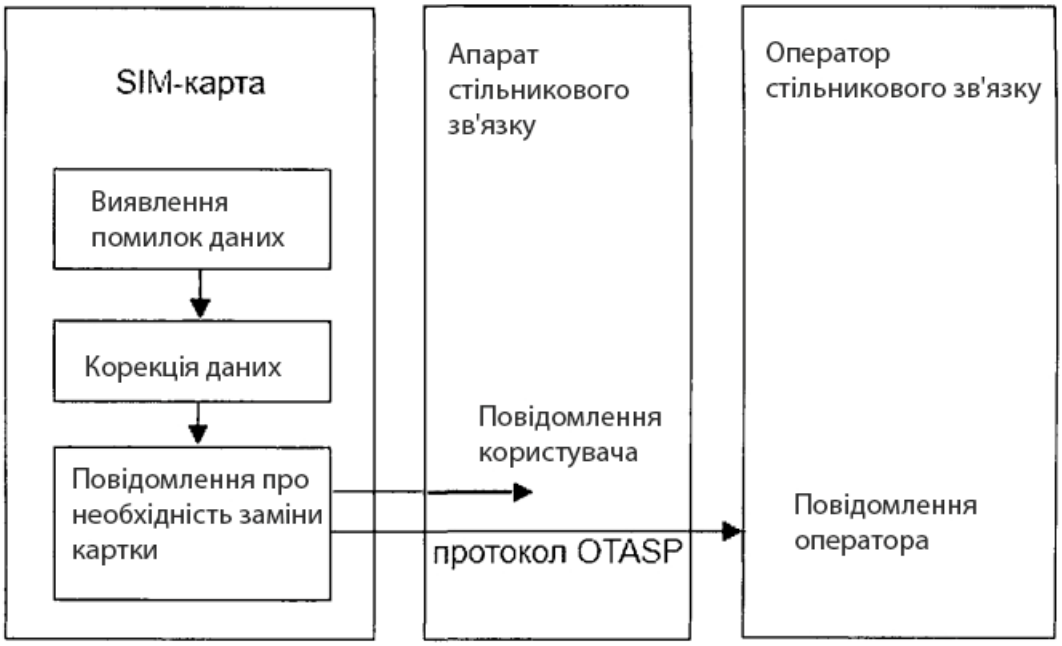

Рис. 3.12. Система захисту від помилок у SIM-картах апаратів стільниковий зв'язку

Таким чином, використання методів завадостійкого кодування дозволить підвищити надійність зберігання даних у SIM-карті, а також виявляти необхідність швидкої заміни SMART-карти. При цьому реалізація алгоритмів завадостійкого кодування в SMART-карті мало позначиться на швидкості її роботи, в силу незначних розмірів даних, що кодуються, і обчислювальних можливостей МКПІ.

3.4.2 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації на базі магнітних карток

У цьому розділі показано, яким чином може підвищитися ефективність роботи МКПІ на базі магнітних карт шляхом застосування розширених методів завадостійкого кодування, розглянутих у другому розділі. Як конкретний приклад розглянемо банківські картки з магнітною смугою.

Банківська карта з магнітною смугою на сьогоднішній день використовується тільки як ідентифікаційний ключ. Вона містить у своїй пам'яті дані про власника, назву банку, номер рахунку.

На відміну від SMART-карт, процедура декодування даних здійснюється на хост-комп'ютері після читання та передачі даних з карти. Тому дані, з якими працює декодер хост-комп'ютера, в загальному випадку містять не тільки помилки даних, що зберігаються, але і помилки, що виникли при передачі даних [105]. Виявити характер помилки, виникла вона під час передачі каналом зв'язку чи у самій карті, неможливо. Отже, виявлення помилок при декодуванні в загальному випадку ще не говорить про несправність карти, а може вказувати на помилки, що з'явилися під час передачі даних на хост-комп'ютер.

У разі виявлення помилки декодер намагається її виправити. Якщо в нього це виходить, здійснюється подальша робота користувача з хост-комп'ютером.

Якщо декодер не може виправити помилки, але при цьому виявляє їх, хост-комп'ютер видає повідомлення про неможливість читання карти, після якого власник картки може самостійно прийняти рішення про необхідність її заміни.

Найменш ймовірною є ситуація, коли в особових даних є помилки, але факт їх наявності декодер виявити не може. У цьому випадку некоректність даних буде виявлена лише при автентифікації користувача.

Робота існуючих на сьогоднішній день МКПІ з магнітними картами схематично представлено рисунку 3.13.

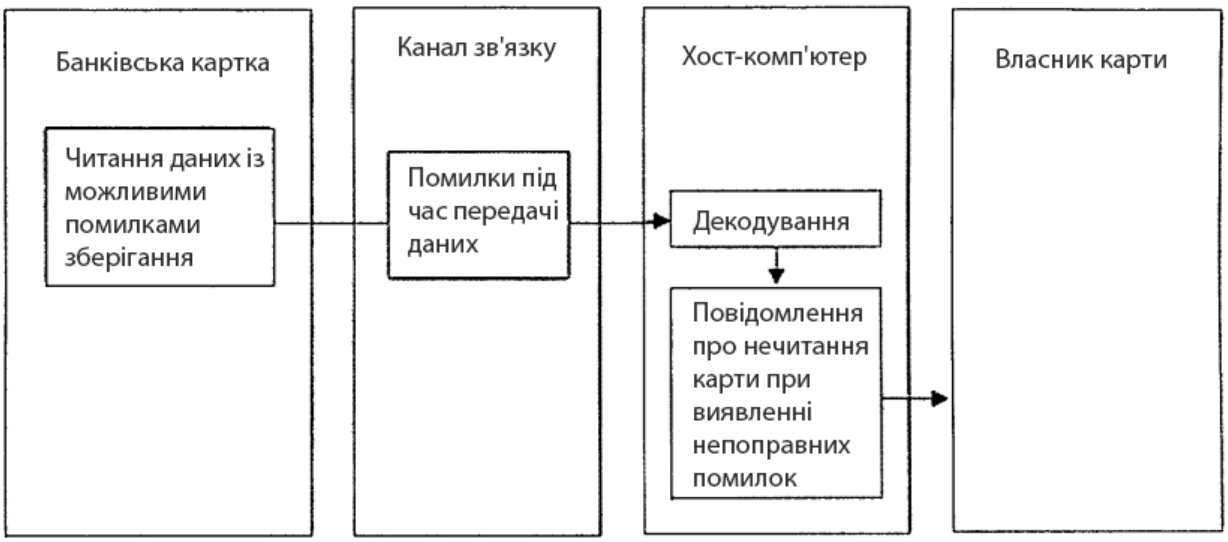

Рис. 3.13. Система захисту від помилок у банківських картках з магнітною смугою

При використанні розширених методів завадостійкого кодування магнітних карт, розглянутих у другому та третьому розділі, схема захисту від помилок буде аналогічна представленій на рисунку 3.13. Відмінності будуть наступними:

- Виникаючі помилки виправлятимуться у більшій кількості випадків. Карта буде «читаною» за наявності більшої кількості помилок, ніж те, що допускається для банківських карток, що використовуються на сьогодні.
- Ймовірність виникнення ситуації, коли в особових даних є помилки, але факт їхньої наявності не виявлено, буде ще меншим, ніж для «звичайних» методів кодування.

Таким чином, застосування розширених методів завадостійкого кодування дозволить підвищити надійність і читання банківських карт з магнітною смугою.

3.4.3 Підвищення надійності мікрокомп'ютерних комплексів персональної ідентифікації на основі технології RFID

Сучасні МКПІ на основі технології RFID використовуються в найрізноманітніших областях, але сфера їх застосування продовжує необмежено зростати. Ефективність багатьох таких МКПІ може бути суттєво

збільшена шляхом застосування методів завадостійкого кодування для даних, що зберігаються в пам'яті RFID-міток.

У цьому розділі розглянемо практичне застосування методів завадостійкого кодування в системах контролю доступу (наприклад, проксіміті-картки або чіп-ключі автомобіля), заснованих на схемах RFID-ідентифікації.

У МКПІ контролю доступу персональним ідентифікатором, що засвідчує особу, є RFID-мітка, що містить ідентифікаційні дані власника та всю історію його переміщень чи змін, списки доступів, привілеїв. При вчиненні відповідних дій власником на згадку RFID-мітки міститься супутня інформація.

Таким чином, модель організації роботи пам'яті RFID-мітки, що використовуються в системах контролю доступу, найбільше нагадують модель послідовного вибору сторінок пам'яті для запису [98].

Відповідними методами завадостійкого кодування для такої моделі роботи пам'яті, розглянутими в другому і третьому розділах, є розбиття даних пам'яті RFID-мітки на кодові блоки та кодування їх кодами Хеммінга або Ріда-Соломона. У міру додавання нових даних у пам'ять RFID-пристрою, вони повинні додаватися до відповідних кодових блоків, або утворювати новий кодовий блок.

Так як при процедурі читання з персонального ідентифікатора може зчитуватись вся історія подій або її значна частина, найбільш доцільним буде розміщення процедур декодування до хост-комп'ютера. Це дозволить уникнути зайвих енерговитрат у RFID-мітці.

Застосування методів завадостійкого кодування в МКПІ контролю доступу на основі технології RFID дозволить збільшити ефективність його функціонування за кількома параметрами:

- Підвищення рівня захищеності даних, що зберігаються у пам'яті RFID-мітки, від помилок. Вищий рівень надійності систем контролю доступу, заснованих на використанні RFID.

- Підвищення довговічності RFID-пристроїв дозволить максимально використовувати їх ресурси працездатності у системах контролю доступу.
- До недоліків методів перешкодостійкого кодування можна віднести деяке збільшення кількості даних, що пересилаються при читанні даних з пам'яті персональних ідентифікаторів.

3.5 Розробка системи динамічного RFID ключа в мікрокомп'ютерних комплексах

У мікрокомп'ютерних комплексах RFID-технологія може використовуватися для різних цілей, наприклад, для:

- Ідентифікації об'єктів.
- Керування доступом.
- Моніторингу стану.

Однак звичайні RFID-мітки мають статичний код, який не можна змінити. Це може призвести до проблем безпеки, оскільки код може бути легко скопійований або підроблений.

Для підвищення безпеки RFID-систем можна використовувати динамічні RFID-ключі. Динамічні RFID-ключі - це коди, які генеруються випадковим чином і оновлюються періодично. Це робить їх більш складними для клонування або підробки.

Принцип роботи динамічного RFID ключа.

Динамічний RFID-ключ складається з двох частин:

- Генератора ключа.
- RFID-мітки.

Генератор ключа генерує випадковий код, який потім записується на RFID-мітку. RFID-мітка містить унікальний ідентифікатор, який також записується на неї. Коли RFID-мітка сканується, генератор ключа генерує новий код. Новий код передається на RFID-мітку, яка замінює старий код.

Розробка системи динамічного RFID ключа в мікрокомп'ютерних комплексах.

Система динамічного RFID ключа в мікрокомп'ютерних комплексах складається з наступних компонентів:

- Мікроконтролер;
- RFID-модуль;
- генератор ключа.

Мікроконтролер використовується для керування системою. RFID-модуль використовується для зчитування даних з RFID-міток. Генератор ключа генерує випадкові коди.

Переваги динамічного RFID ключа:

- Більш висока безпека.
- Менша ймовірність клонування або підробки.
- Покращена конфіденційність.

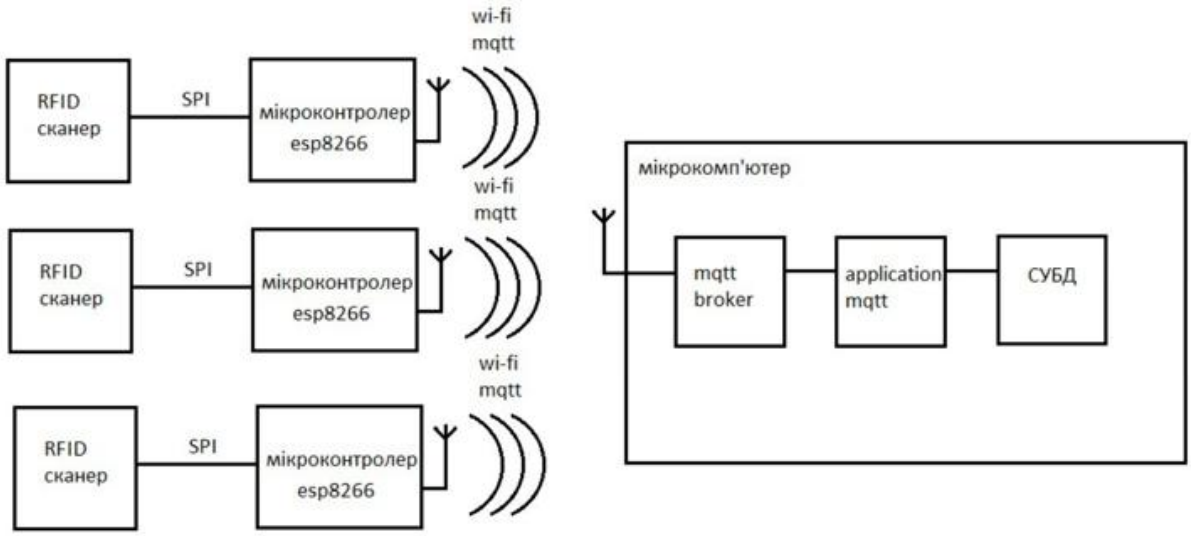

Рис. 3.14 Система ідентифікації RFID з мікрокомп'ютером

Дана схема показує систему, яка використовує мікроконтролер ESP8266 для читання даних з RFID-сканера і передачі їх на MQTT-брокер. Дані MQTT потім можуть бути отримані додатком або СУБД.

Мікрокомп'ютер ESP8266 є недорогим і енергоефективним пристроєм, який має вбудований Wi-Fi. Він ідеально підходить для таких застосувань, як цей, де потрібно надсилати дані через Інтернет [95].

RFID-сканер використовується для зчитування даних з RFID-міток. Мітки RFID є невеликими пристроями, які містять унікальний код. Цей код може використовуватися для ідентифікації об'єктів або осіб.

MQTT-брокер - це програма, яка приймає та пересилає повідомлення MQTT. MQTT - це протокол передачі повідомлень, який часто використовується в Інтернеті речей (IoT).

Додаток або СУБД можуть використовуватися для отримання даних, отриманих від MQTT-брокера. Додаток може використовувати ці дані для виконання певних дій, наприклад, для керування обладнанням або для зберігання даних.

Систему можна розділити на кілька модулів:

- модуль RFID відповідає за зчитування даних з RFID-міток;
- модуль мікроконтролера відповідає за обробку даних, отриманих від модуля RFID, і за передачу їх на MQTT-брокер;
- модуль MQTT відповідає за передачу даних на MQTT-брокер;
- модуль додатку або СУБД відповідає за отримання даних від MQTT-брокера;

Цей поділ дозволяє розробникам системи модифікувати або замінювати окремі модулі без впливу на інші модулі. Наприклад, розробник може замінити RFID-сканер на інший сканер або замінити додаток або СУБД на інший додаток або СУБД.

В результаті динамічні RFID-ключі є ефективним способом підвищення безпеки RFID-систем.

Висновки до третього розділу

Третій розділ досліджень присвячений модифікації МКПІ для вдосконалення теоретичної та технічної бази, що дозволяють підвищити

надійність кожної групи МКПІ та захищеність даних, що зберігаються в них, від помилок:

- Розроблені модифікації файлової системи МКПІ дозволяють підвищити надійність та захищеність даних, що зберігаються в них, за рахунок використання кодів Хеммінга, Ріда-Соломона та локального коду надлишковості для виявлення та виправлення помилок у даних. Це дозволяє відновити дані, пошкоджені в результаті шуму або зловмисних дій. Використання методів динамічного управління сторінками пам'яті дозволяє зменшити кількість сторінок пам'яті, які необхідно перевіряти на наявність помилок, що підвищує ефективність роботи файлової системи. Використання методів резервування даних дозволяє відновити дані, якщо вони повністю пошкоджені.
- Розроблено метод обробки та введення-виведення даних, що дозволяє підвищити надійність та захищеність даних, що зберігаються в МКПІ, за рахунок використання методів синхронізації даних. Це дозволяє забезпечити збереження даних у разі збоїв у роботі МКПІ. Використання методів захисту даних від несанкціонованого доступу дозволяє запобігти доступу до даних зловмисниками. Використання методів підвищення ефективності обробки даних дозволяє зменшити кількість обчислень, що виконує МКПІ, що знижує ризик виникнення помилок.
- Розроблена система динамічного RFID ключа в мікрокомп'ютерних комплексах дозволяє підвищити надійність та захищеність даних, що зберігаються в них, за рахунок використання біометричних даних для генерації ключа. Це ускладнює для зловмисників отримання ключа. Ключ змінюється кожний раз, коли RFID-мітка потрапляє в зону дії RFID-сканера, це ускладнює для зловмисників отримання доступу до даних, навіть якщо вони знають попередній ключ.

Розроблені модифікації для вдосконалення теоретичної та технічної бази МКПІ дозволяють підвищити надійність та захищеність даних, що зберігаються в них. Ці модифікації можуть бути використані для підвищення

надійності та захищеності даних у різних групах МКПІ, включаючи SMART-карти, RFID-пристрої та біометричні паспорти.

ВИСНОВКИ

У ході дисертаційного дослідження було розроблено та впроваджено методи обробки даних в мікрокомп’ютерних комплексах на основі персональної ідентифікації.

Проаналізовано особливості функціонування та організації зберігання інформації в персональних ідентифікаторах МКПІ, а також можливості підвищення надійності шляхом модифікації обробки, зберігання та введення-виведення даних та аналізу помилок, характерних для МКПІ. Здійснено аналіз інформації про наявні мікрокомп'ютерні системи, які використовуються для особистої ідентифікації. Представлено архітектуру та особливості зберігання та розподілу інформації в RFID-мітках з функцією пам'яті. Представлено групу умов, яким повинні задовольняти алгоритми та методи підвищення надійності, що застосовуються у SMART-картах, Наведені можливості підвищення надійності магнітних карток. Проаналізовані помилки, які виникають у мікрокомп'ютерних комплексах персональної ідентифікації. Представлені рекомендації щодо підвищення надійності МКПІ. Надані пропозиції щодо змін до базових стандартів для SMART-карт, магнітних карт, біометричних паспортів та RFID-систем, що регламентують обробку, зберігання та введення-виведення даних.

В процесі досліджень було:

- розроблено модель для випадкового та послідовного вибору сторінок для введення-виводу даних, що дозволяє оцінювати алгоритми завадостійкого кодування інформації в МКПІ з позицій формул для розрахунку ефективності завадостійкого коду, коефіцієнта витрати пам'яті та ступеня захищеності даних;

- запропоновано методи організації введення-виводу та зберігання даних, що забезпечують надійність SMART-карт, RFID-пристроїв та біометричних паспортів;

- розглянуто застосування кодів Хеммінга, Ріда-Соломона та локального коду надлишковості в моделі випадкового та послідовного вибору сторінок для введення-виведення даних. Проведено розрахунки складності операцій кодування та декодування даних у персональні ідентифікатори. Вибрано найефективніші модифікації алгоритмів для зберігання персональних даних в ЕСППЗП SMART-карт, RFID-пристроїв, біометричних паспортів та магнітних карт;

- розроблені модифікації файлової системи МКПІ, що дозволяють підвищити надійність та захищеність даних, що зберігаються в них, за рахунок використання кодів Хеммінга, Ріда-Соломона та поздовжнього контролю надмірності для виявлення та виправлення помилок у даних. Це дозволяє відновити дані, пошкоджені в результаті шуму або зловмисних дій;

- розроблено метод обробки та введення-виведення даних, що дозволяє підвищити надійність та захищеність даних, що зберігаються в МКПІ, за рахунок використання методів синхронізації даних. Це дозволяє забезпечити збереження даних у разі збоїв у роботі МКПІ;

- розроблена система динамічного RFID ключа в мікрокомп'ютерних комплексах дозволяє підвищити надійність та захищеність даних, що зберігаються в них, за рахунок використання біометричних даних для генерації ключа.

Результати наукових досліджень були використані на кафедрі Інженерії програмного забезпечення автоматизованих систем навчально-наукового інституту Інформаційних технологій Державного університету інформаційно-комунікаційних технологій під час виконання науково-дослідної роботи на тему: «Дослідження обробки траєкторної інформації в вимірювально-обчислювальних системах» (ДУІКТ, м. Київ), впроваджені у виробничий процес на підприємстві ТОВ «Хуавей» Україна, для підвищення надійності персональних пристроїв на базі SMART-карт, що використовуються для аутентифікації користувачів. Впровадження дозволило підвищити надійність таких пристроїв на 23%, а в 89% випадків

відновити дані в разі виникнення помилок, та в навчальному процесі Державного університету інформаційно-комунікаційних технологій (Київ).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Danylchenko V. M., Sribna I. M. Development of computer barcode detector based on one chip microcomputer. Connectivity. 2021. Vol. 151, no. http://con.dut.edu.ua/index.php/communication/article/view/2537

2. Danylchenko V.M. Mykolaichuk V.R. Improving the reliability of storing information in microcomputer complexes of personal identification. XI міжнародна науково-технічної конференції студентства «Світ інформації та телекомунікації». Київ, 2021 року – 218-220 с. http://www.dut.edu.ua/uploads/n_9179_17683450.pdf

3. В.М. Чорна, О. М. Ткаленко, О. В. Полоневич, О. В. Сеньков, Особливості захисту інформації в NFC, Науковий журнал “Зв’язок”, №4, с. 25-29, 2018. http://con.dut.edu.ua/index.php/communication/article/view/2118

4. К. П. Сторчак, А. М. Тушич, О.М. Ткаленко, Т.М. Жила, В.М. Чорна, Аналіз вимог до проектування хмарної платформи для Інтернету речей. Науковий журнал “Зв’язок”, №4, с. 30-35, 2019. http://con.dut.edu.ua/index.php/communication/article/view/2378

5. В.М. Данильченко, Дослідження мобільної онлайн системи блокування мікрокомп’ютера на основі технології NFC, Науковий журнал “Зв’язок”, №5, с. 20-24, 2023. https://doi.org/10.31673/2412-9070.2023.052022

6. Analysis of requirements for designing a cloud platform for the Internet of things / K. P. Storchak та ін. Connectivity. 2019. T. 142, № 6. C. 8–11. https://con.dut.edu.ua/index.php/communication/article/view/2378

7. Wang M S, Guo Y X, Wu W. Near-field and far-field shared structure for NF Cand CNSS applications [J]. Iet Microwaves Antennas & Propagation, 2020, 11(15)

8. Wu D Z. Design and Application for Five Anti-wireless System on Beijing Taiyanggong Power Plant [D]. North China Electric Power University, 2019

9. Павлиш В. А., Гліненко Л. К., Шаховська Н. Б. Основи інформаційних технологій і ситем. 2019

10. Congwei Hu, Wu Chen, Shan Gao. Data Pro-cessing For GPS Precise Point Positioning // Trans-actions of Nanjing University of Aeronautics & As-tronautics. 2020. 22(2).

11. Матвієнко М.П., Розен В.П., Закладний О.М. Архітектура комп'ютера. 2019

12. Shariar Makarechi, Research Methodology for Building Automation Performance Index / Shariar Makarechi, Roozbeh Kangari // International Journal of Facility Management. – 2019. - №1, vol.2, P. 886 – 910.

13. Anatolii Syrotenko, Evaluation of Quality of Hidden Transmitters Detection by a Group of Coordinated Unmanned Aerial Vehicles on the basis of Entropy Decrease Dynamics / Pavlo Shchypanskyi, Vitalii Savchenko, Denis Krashchenko, Kseniia Yerhidzei, Ivan Havryliuk // International Journal of Emerging Trends in Engineering Research. - 2020. – No. 6, Volume 8, P. 2798-2804.

14. Henrik Dibowski, Automated Design of Building Automation Systems / Henrik Dibowski, Joern Ploennigs// IEEE Transactions on Industrial Electronics. – 2019. - №11, P. 3606 – 3613.

15. Lewandowski, M.; Płaczek, B. Data Transmission Reduction in Wireless Sensor Network for Spatial Event Detection. Sensors 2021, 21, 7256. https://doi.org/10.3390/s21217256

16. Zhu, X. Complex event detection for commodity distribution Internet of Things model incorporating radio frequency identification and Wireless Sensor Network. Future Gener. Comput. Syst. 2021, 125, 100–111. https://doi.org/10.1016/j.future.2021.06.024

17. J.M. Williams, R. Khanna, J.P. Ruiz-Rosero, et al., Weaving the wireless web: toward a low-power, dense wireless sensor network for the industrial IoT, IEEE Microw. Mag. 18 (7) (2019) 40–63

18. Тимченко О. Особливості маршрутизації в безпроводових сенсорних мережах [Електронний ресурс] / О. Тимченко, М. Зеляновський // Українська академія друкарства 2021

19. Я. О. Галай, А. П. Бондарчук, О. М. Ткаленко, О. В. Полоневич, О. В. Зінченко. Методика забезпечення захисту корпоративної мережі зв'язку при віддаленому управлінні / Зв'язок. – 2021. – №1. – С. 55-58.

20. The adjacency matrix-based algorithm of constructing barrier coverage in underwater wireless sensor network [Електронний ресурс] Juan Chang // IEEE Xplore – Режим доступу: https://ieeexplore.ieee.org/document/8242477 (дата звернення 02.09.2021).

Louati, A., Louati, H. and Li, Z. 'Deep learning and case-based reasoning for predictive and adaptive traffic emergency management', Journal of Supercomputing, 77(5), pp. 4389–4418. https://doi.org/10.1007/s11227-020-03435-3.

21. Mahmoud, A. M. A. et al. 'Monitoring the Dynamics of Formby Sand Dunes Using Airborne LiDAR DTMs', Remote Sensing, 13(22), p. 4665. https://doi.org/10.3390/rs13224665.

22. Maini, P., Tokekar, P. and Sujit, P. B. 'Visibility-Based Persistent Monitoring of Piecewise Linear Features on a Terrain Using Multiple Aerial and Ground Robots', IEEE Transactions on Automation Science & Engineering, 18(4), pp. 1692–1704. doi:10.1109/TASE.2020.3014949.

23. Malik, H. et al. 'Development of wide area monitoring system for smart grid application', Journal of Intelligent & Fuzzy Systems, pp. 1–13. https://doi.org/10.3233/jifs-189736.

24. Технології інтернету речей. Навчальний посібник [Електронний ресурс] Б.Ю. Жураковський, І.О. Зенів // КПІ ім. Ігоря Сікорського – Режим 225 доступу: https://ela.kpi.ua/bitstream/123456789/42078 (дата звернення 02.09.2021).

25. Overcoming Bandwidth Limitations in Wireless Sensor Networks by Exploitation of Cyclic Signal Patterns: An Event-triggered Learning Approach

[Електронний ресурс] Jonas Beuchert et al. // MDPI – Режим доступу: https://www.mdpi.com/1424-8220/20/1/260 (дата звернення 02.09.2021)

26. A Systematic Review of Quality of Service in Wireless Sensor Networks using Machine Learning: Recent Trend and Future Vision [Електронний ресурс] Meena Pundir // Journal of Network and Computer Applications – Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S1084804521001065 (дата звернення 02.09.2021).

27. A Qualitative evaluation of the process of functionally stable recovery control of the aircraft in emergencies with an algorithm based on solving inverse dynamic problems [Електронний ресурс] Oleg Mashkov et al. // CEUR Workshop Proceedings–Режим доступу: http://ceur-ws.org/Vol-2631/paper29.pdf (дата звернення 02.09.2021).

28. Smart Home Technology Network Construction Method and Device Interaction organization Concept [Електронний ресурс] Viktor Vyshnivskyi et al. // 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC) – Режим доступу: https://ieeexplore.ieee.org/document/9239220/authors#authors (дата звернення 02.09.2021).

29. О. В. Семко, В. В. Семко. Розробка формальної моделі інтелектуального управління маршрутизацією в конфліктуючих сенсорних мережах варіативної топології // Математичне моделювання в економіці. - 2019. - № 1. - С. 5- 19.

30. Wireless Sensor Networks Formation: Approaches and Techniques [Електронний ресурс] Miriam Carlos-Mancilla, Ernesto López-Mellado, Mario Siller // – Режим доступу: https://www.hindawi.com/journals/js/2016/2081902 / (дата звернення 02.09.2021).

31. Song, Z. et al. ‘Deep reinforcement learning for permanent magnet synchronous motor speed control systems’, Neural Computing & Applications, 33(10), pp. 5409–5418. https://doi.org/10.1007/s00521-020-05352-1.

32. Stiros SC. GNSS (GPS) Monitoring of Dynamic Deflections of Bridges: Structural Constraints and Metrological Limitations. Infrastructures. 2021; 6(2):23. https://doi.org/10.3390/infrastructures6020023

33. Shariar Makarechi, Research Methodology for Building Automation Performance Index / Shariar Makarechi, Roozbeh Kangari // International Journal of Facility Management. – 2015. - №1, vol.2, P. 886 – 910.

34. Sun, Z., Chen, P. and Zhang, X. 'An Intelligent Optimization Strategy Based on Deep Reinforcement Learning for Step Counting', Discrete Dynamics in Nature & Society, pp. 1–10. https://doi.org/10.1155/2021/9536309.

35. Шрам М.М. А.В. Лемешко, О.М. Ткаченко, Д.В. Сорокін, Д.В. Кращенко Оброблення даних кінцевими пристроями в системах IoT (Інтернет речей) / Зв'язок. - 2021. - №5, С. 24-28.

36. Vaz, R. et al. 'Event detection and classification through wavelet-based method in low voltage wide-area monitoring systems', International Journal of Electrical Power & Energy Systems, 130, p. N.PAG. https://doi.org/10.1016/j.ijepes.2021.106919.

37. Wu, Y. et al. 'Dynamic handoff policy for RAN slicing by exploiting deep reinforcement learning', EURASIP Journal on Wireless Communications & Networking, 2021(1), pp. 1–17. https://doi.org/10.1186/s13638-021-01939-x.

38. Ali, B. et al. 'Reinforcement learning based metric filtering for evolutionary distance metric learning', Intelligent Data Analysis, 24(6), pp. 1345–1364. doi: 10.3233/IDA-194887.

39. Bhamidipati, S. et al. . Artificial-Intelligence-Based Distributed Belief Propagation and Recurrent Neural Network Algorithm for Wide-Area Monitoring Systems', IEEE Network, 34(3), pp. 64–72. https://doi.org/10.1109/MNET.011.1900322.

40. Bondarchuk A., Vyshnivskyi V. Methods and algorithms of optimal controls based on quasistationary adaptation processes / at al // Danish Scientific Journal (DSJ) Denmark. 2020. №40. PP. 53– 59.

41. Atar R., Dupuis P. Characterization of the value function for a differential game formulation of a queueing network optimization problem. [Електронний ресурс] // 1999 Conference on Decision and Control, Phoenix, AZ, USA https://doi.org/10.1109/cdc.1999.832763

42. Chen, Z. and Wang, X. 'Decentralized computation offloading for multi-user mobile edge computing: a deep reinforcement learning approach', EURASIP Journal on Wireless Communications & Networking, 2020(1), p. N.PAG. https://doi.org/10.1186/s13638-020-01801-6.

43. Ching-Sheng Lin et al. 'Empirical explorations of strategic reinforcement learning: a case study in the sorting problem', Proceedings of the Estonian Academy of Sciences, 69(3), pp. 186–196. https://doi.org/10.3176/proc.2020.3.02.

44. Daryanavard, S. and Porr, B. 'Closed-Loop Deep Learning: Generating Forward Models With Backpropagation', Neural Computation, 32(11), pp. 2122–2144. https://doi.org/10.1162/neco_a_01317.

45. Duan, W. et al. 'Automatic alignment of contemporary vector data and georeferenced historical maps using reinforcement learning', International Journal of Geographical Information Science, 34(4), pp. 824–849. https://doi.org/10.1080/13658816.2019.1698742.

46. Editorial Intelligent Data Analysis, 24(6), pp. 1235–1237. https://doi.org/10.3233/IDA-200016.

47. Espinosa-Leal, L. et al. 'Autonomous Industrial Management via Reinforcement Learning', Journal of Intelligent & Fuzzy Systems, 39(6), pp. 8427–8439. https://doi.org/10.3233/JIFS-189161.

48. Han, R., Chen, K. and Tan, C. 'Curiosity-driven recommendation strategy for adaptive learning via deep reinforcement learning', British Journal of Mathematical & Statistical Psychology, 73(3), pp. 522–540. https://doi.org/10.1111/bmsp.12199.

49. Hartl, R., Hansjakob, J. & Zaeh, M.F. Improving the surface quality of friction stir welds using reinforcement learning and Bayesian optimization. Int J

Adv Manuf Technol 110, 3145–3167 . https://doi.org/10.1007/s00170-020-05696-x.

50. IoT vs AI: Difference Between Internet of Things and Artificial Intelligence [Електронний ресурс] // – Режим доступу: https://www.upgrad.com/blog/iot-vs-ai/ (дата звернення 02.09.2021).

51. Толубко В. Б., Беркман Л. Н., Заїка В. Ф., Варфоломеєва О. Г., Домрачева К. О. Метод аналізу якості функціонування системи керування мережею наступного покоління на основі інформаційно-ентропійних характеристик // Зв'язок. - 2019. - №6, С. 3-8

52. Jaunet, T., Vuillemot, R. and Wolf, C. 'DRLViz: Understanding Decisions and Memory in Deep Reinforcement Learning', Computer Graphics Forum, 39(3), pp. 49–61. https://doi.org/10.1111/cgf.13962.

53. Kinose, A. and Taniguchi, T. 'Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model', Advanced Robotics, 34(16), pp. 1055–1067. https://doi.org/10.1080/01691864.2020.1778521.

54. Leem, J. and Kim, H. Y. 'Action-specialized expert ensemble trading system with extended discrete action space using deep reinforcement learning', PLoS ONE, 15(7), pp. 1–39. https://doi.org/10.1371/journal.pone.0236178.

55. Li, W. and Hsu, C.-Y. 'Automated terrain feature identification from remote sensing imagery: a deep learning approach', International Journal of Geographical Information Science, 34(4), pp. 637–660. https://doi.org/10.1080/13658816.2018.1542697.

AL-Hawawreh, M., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on Deep Learning Models. Journal of Information Security and Applications, 41, 1–11. https://doi.org/10.1016/j.jisa.2018.05.002.

57. A K, H. (2021). Techniques for detecting malware traffic: A comprehensive approach to feature selection and classification. International

Journal for Research in Applied Science and Engineering Technology, 9(12), 1–10. https://doi.org/10.22214/ijraset.2021.39088.

58. Juliy Mikolayovych Boiko, Alexander Ivanovych Eromenko. Improvements Encoding Energy Benefit in Protected Telecommunication Data Transmission Channels. Scientific Journal «Communications». Science Publishing Group, USA. Vol. 2, No. 1, 2014, pp. 7–14. https://doi.org/10.11648/j.com.20140201.12.

59. Paxson, V., Allman, M., Dawson, S., Fenner, W., Griner, J., Heavens, I., Lahey, K., Semke, J., & Volz, B. (1999). Known TCP implementation problems. https://doi.org/10.17487/rfc2525.

60. Boiko J. M., Eromenko A. I. Synthesis problems of clock synchronization devices for receivers of satellite telecommunication data transmission systems. Bulletin of National Technical University of Ukraine "Kyiv Polytechnic Institute" Series — Radiotechnique. Radioapparatus building. 2014. Ed. 58. 55–66 p.

61. Sorokin, I. (2011). Comparing files using structural entropy. Journal in Computer Virology, 7(4), 259–265. https://doi.org/10.1007/s11416-011-0153-9.

62. Бойко Ю. М. Оцінювання якісних показників пристроїв синхронізації сигналів засобів телекомунікацій. Вісник Хмельницького національного університету. 2015. № 1. С. 204–213.

63. M. A. Platas-Garza and J. A. de la O Serna, "Polynomial Implementation of the Taylor–Fourier Transform for Harmonic Analysis," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 12, pp. 2846-2854, Dec. 2014, https://doi.org/10.1109/TIM.2014.2324191.

64. Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy. https://doi.org/10.5220/0006639801080116.

65. Телекомунікаційні системи та мережі. Структура й основні функції. Том 1. / Поповський В.В., Лемешко О.В., Ковальчук В.К., Плотніков

М.Д., Картушин Ю.П. та інші. [Електронний ресурс]. – Х.: ХНУРЕ. 2018. Режим доступу – http://www.znanius.com/3534.html

66. M. Ficco. Hybrid indoor and outdoor location services for new generation mobile terminals / M. Ficco, F. Palmieri, A. Castiglione // Personal and Ubiquitous Computing - 2014 - №12, pp. 271–285.

67. Luo, T.-L. et al. 'A framework of deep reinforcement learning for stock evaluation functions', Journal of Intelligent & Fuzzy Systems, 38(5), pp. 5639–5649. https://doi.org/10.3233/JIFS-179653.

68. Ma, X. and Zhou, Q. 'Special issue on deep learning and neural computing for intelligent sensing and control', Neural Computing & Applications, 32(7), pp. 1811–1812. https://doi.org/10.1007/s00521-020-04785-y.

69. О.Л.Туровський «Проблеми та особливості оцінки частоти сигналу, що приймається супутниковою системою зв'язку», Проблеми інформатізації: IV міжнар. наук.-техн. конф., Київ-Полтава-Катовіце-ПарижОрел-Харьків, 09-10 квітня 2015 р., С. 39-40.

70. Najar, A. et al. 'The actions of others act as a pseudo-reward to drive imitation in the context of social reinforcement learning', PLoS Biology, 18(12), pp. 1–25. https://doi.org/10.1371/journal.pbio.3001028.

71. Álvaro Villalba. servIoTicy and iServe: A Scalable Platform for Mining the IoT / Álvaro Villalba // The 6th International Conference on Ambient Systems, Networks and Technologies – 2015 - pp. 1022 –1027.

72. Passalis, N. and Tefas, A. 'Continuous drone control using deep reinforcement learning for frontal view person shooting', Neural Computing & Applications, 32(9), pp. 4227–4238. https://doi.org/10.1007/s00521-019-04330-6.

73. Hui Suo. Security in the Internet of Things: A Review / Hui Suo // International Conference on Computer Science and Electronics Engineering, ICCSEE - 2022 - pp. 2022–2024.

74. Luka Milić. A novel versatile architecture for Internet of Things / Luka Milić, Leonardo Jelenković // 38th International Convention on Information

and Communica- tion Technology, Electronics and Microelectronics,– 2015 - pp. 1026– 1031.

75. Sethi, K. et al. ‘A context-aware robust intrusion detection system: a reinforcement learning-based approach’, International Journal of Information Security, 19(6), pp. 657–678. https://doi.org/10.1007/s10207-019-00482-7.

76. Shahbazi, Z. and Byun, Y.-C. ‘Topic modeling in short-text using non-negative matrix factorization based on deep reinforcement learning’, Journal of Intelligent & Fuzzy Systems, 39(1), pp. 753–770. https://doi.org/10.3233/JIFS-191690.

77. Pasquale Foggia, Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion / Pasquale Foggia, Alessia Saggese // IEEE Transactions on Circuits and Systems for Video Technology. – 2015. - №1, P. 1267-1281.

78. Johnny K.W. Wong, Development of intelligence analytic models for integrated building management systems (IBMS) in intelligent buildings / Johnny K.W. Wong, Heng Li // Intelligent Buildings International. – 2009. – №1, P. 5-22.

79. Кузьмін О.В. Оптимізація кластерної структури сенсорної мережі методом імітаційного моделювання / О.В. Кузьмін, В.А. Головко // Комп'ютерні науки та інформаційні технології. – 2010. – №47. – №1. – С. 103-107.

80. Бондарчук, А. П., Г. С. Срочинська, and М. Г. Твердохліб. "Основи інфокомунікаційних технологій. Навчальний посібник [Електронний ресурс]." Державний університет теле-комунікацій, Київ.– 2015.–76 с.–Режим доступу: http://www.dut. edu. ua/uploads/l_840_37756081. pdf 5 (2015).

81. Zhihong Yang. Study and application on the architecture and key technologies for IOT / Zhihong Yang // International Conference on Multimedia Technology. –2011 – pp. 747–751.

82. Sleaman, W. K. et al. ‘Indoor mobile robot navigation using deep convolutional neural network’, Journal of Intelligent & Fuzzy Systems, 39(4), pp. 5475–5486. https://doi.org/10.3233/JIFS-189030.

83. Sun, S. et al. ‘Inverse reinforcement learning-based time-dependent A* planner for human-aware robot navigation with local vision’, Advanced Robotics, 34(13), pp. 888–901. https://doi.org/10.1080/01691864.2020.1753569.

84. Стеклов В. К., Беркман Л. Н., Кільчицький Є. В. Оптимізація та моделювання пристроїв і систем зв’язку. - К.: Техніка, 2004. - 576 с.

85. Jaehak Yu. IoT as a applications: cloud-based building management systems for the internet of things / Jaehak Yu // Multimedia Tools and Applications 75.22 - 2016 - , pp. 14583– 14596.

86. Taniguchi, A. et al. ‘Spatial concept-based navigation with human speech instructions via probabilistic inference on Bayesian generative model’, Advanced Robotics, 34(19), pp. 1213–1228. https://doi.org/10.1080/01691864.2020.1817777.

87. Treesatayapun, C. ‘Knowledge-based reinforcement learning controller with fuzzy-rule network: experimental validation’, Neural Computing & Applications, 32(13), pp. 9761–9775. https://doi.org/10.1007/s00521-019-04509-x.

88. Stryczek, S., & Natkaniec, M. (2022). Internet threat detection in smart grids based on network traffic analysis using LSTM, if, and SVM. Energies, 16(1), 329. https://doi.org/10.3390/en16010329.

89. Yoon, J. et al. ‘Design of reinforcement learning for perimeter control using network transmission model based macroscopic traffic simulation’, PLoS ONE, 15(7), pp. 1–16. https://doi.org/10.1371/journal.pone.0236655.

90. Martin Bauer. Final Architectural Reference Model for IoT /Martin Bauer [Електронний ресурс] // – Режим доступу: http://www.iot-a.eu – 2013 - pp. 53–59 (дата звернення 04.09.2021).

91. R. Khan. Future Internet: The Internet of Things Architecture, Possible Appli- cations and Key Challenges / R. Khan // 10th International Conference on Frontiers of Information Technology. – 2012 – pp. 257–260.

92. I. Yaqoob. Internet of Things Architecture: Recent Advances, Taxonomy, Re- quirements, and Open Challenges/ I. Yaqoob // IEEE Wireless Communications 24.3 –2017 – pp. 10–16.

93. Zhang, T. et al. 'Robotic constant-force grinding control with a press-and-release model and model-based reinforcement learning', International Journal of Advanced Manufacturing Technology, 106(1/2), pp. 589–602. https://doi.org/10.1007/s00170-019-04614-0.

94. Bucci, M. A. et al. 'Control of chaotic systems by deep reinforcement learning', Proceedings of the Royal Society A: Mathematical, Physical & Engineering Sciences, 475(2231), pp. 1–20. https://doi.org/10.1098/rspa.2019.0351.

95. Коцюбівська К., Прісич В., Яворський О. Впровадження технологій інтернету речей під час створення системи «Розумний дім». Цифрова платформа. // Інформаційні технології в соціокультурній сфері. - 2019 - Т. 2, № 2., С. 136–143.

96. Cyr, A. and Thériault, F. 'Spatial Concept Learning: A Spiking Neural Network Implementation in Virtual and Physical Robots', Computational Intelligence & Neuroscience, pp. 1–8. https://doi.org/10.1155/2019/8361369.

97. Базилевич, В. М., Мальцева, М. В., Петренко, Т. А., Черниш, Л. Г. Захищена система розумного будинку з використанням Internet of Things. // Технічні науки та технології, - 2020 - №2, С.218–228.

98. Ha, D. 'Reinforcement Learning for Improving Agent Design', Artificial Life, 25(4), pp. 352–365. https://doi.org/10.1162/artl_a_00301.

99. J. C. Jesus, J. A. Bottega, M. A. S. L. Cuadros and D. F. T. Gamarra 'Deep Deterministic Policy Gradient for Navigation of Mobile Robots in Simulated Environments', 19th International Conference on Advanced Robotics (ICAR), 2019, pp. 362-367, https://doi.org/10.1109/ICAR46387.2019.8981638.

100. Zhihong Yang. Study and application on the architecture and key technologies for IOT / Zhihong Yang // International Conference on Multimedia Technology. –2011 – pp. 747–751.

101. Лаптєв, С., Лаптєва, Т., Гребенніков, А., Кітура, О., Марченко, В., & Луценко, М. (2021). Удосконалення моделі протидії витоку інсайдерської інформації з урахуванням нестабільності функціонування окремих елементів системи захисту інформації. Journal of Science. Lyon, (20/2021). ISSN 3475- 3281.

102. Стеклов В.К., Беркман Л.Н., Костік Б.Я., Олійник В.Ф., Скляренко С.М. та інші. Управління телекомунікаціями із застосуванням новітніх технологій – / Підручник для ВНЗ.К.: Техніка, 2007.- 384 с.

103. Tupitsin, N., & Mykolaichuk, V. System of UAV Failure Modeling for External Pilot Training in Emergency Situations', 2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2019, pp. 57-60, https://doi.org/10.1109/APUAVD47061.2019.8943838.

104. Sotiris Papantoniou. Using Intelligent Building Energy Management System for the Integration of Several Systems to one Overall Monitoring and Management System /Sotiris Papantoniou, Stefano Mangili, and Ivan Mangialenti // 8th International Conference on Sustainability in Energy and Buildings, – September 2016 – Turin, Italy– pp. 639 –647.

105. Luigi Atzori. The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization./ Luigi Atzori // Computer Networks - №56 – 2012 – pp. 3594 – 3608.

106. Wei, S., Jin, X. and Li, H. 'General solutions for nonlinear differential equations: a rule-based self-learning approach using deep reinforcement learning', Computational Mechanics, 64(5), pp. 1361–1374. https://doi.org/10.1007/s00466-019-01715-1.

107. Opirskyy I., Проблеми та загрози безпеці IoT пристроїв. / Holovchak R., Moisiichuk I., Balianda T., & Haraniuk S // Електронне фахове наукове видання «Кібербезпека: освіта, наука, » - 2021 - №3(11), С. 31-42.

108. Жураковський Б. Ю, Зенів І. О. Технології Інтернету Речей – / Навчальний посібник для ВНЗ.К.: КПІ ім. Сікорського, 2021.- 271 с.

ДОДАТОК А

Вихідний код який перетворює запити від ESP на запит в базу та формує відповідь з новим ключем.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "XXXXXX";
const char* password = "xxxxxxx";
const char* mqtt_server = "xxx.xxx.x.xx";

WiFiClient espClient;
PubSubClient client(espClient);

char msg[17];
int value = 0;

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         D1
#define SS_PIN          D2

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup_wifi() {

  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
```

```
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {

  MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance
  mfrc522.PCD_Init();        // Init MFRC522 card
  byte buffer[18];

  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  Serial.print(F(" PICC type: "));
  MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
  Serial.println(mfrc522.PICC_GetTypeName(piccType));

  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  byte block;
  MFRC522::StatusCode status;
  byte len;

  block = 1;

  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1,
&key, &(mfrc522.uid));
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

  status = mfrc522.MIFARE_Read(block, buffer, &len);
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
```

```
if (buffer[0] == payload[16] &&
   buffer[1] == payload[17] &&
   buffer[2] == payload[18] &&
   buffer[3] == payload[19] &&
   buffer[4] == payload[20] &&
   buffer[5] == payload[21] &&
   buffer[6] == payload[22] &&
   buffer[7] == payload[23] &&
   buffer[8] == payload[24] &&
   buffer[9] == payload[25] &&
   buffer[10] == payload[26] &&
   buffer[11] == payload[27] &&
   buffer[12] == payload[28] &&
   buffer[13] == payload[29] &&
   buffer[14] == payload[30] &&
   buffer[15] == payload[31] )
 {
   //WRITE BLOCK

    block = 1;
    //Serial.println(F("Authenticating using key A...")); 
    status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
block, &key, &(mfrc522.uid));
    if (status != MFRC522::STATUS_OK) {
      Serial.print(F("PCD_Authenticate() failed: "));
      Serial.println(mfrc522.GetStatusCodeName(status));
      return;
    }
    else Serial.println(F("PCD_Authenticate() success: "));

    // Write block
    status = mfrc522.MIFARE_Write(block, payload, 16);
    if (status != MFRC522::STATUS_OK) {
      Serial.print(F("MIFARE_Write() failed: "));
      Serial.println(mfrc522.GetStatusCodeName(status));
      return;
    }
    else {
     Serial.println(F("MIFARE_Write() success: "));
       //PRINT BLOCK
     for (uint8_t i = 0; i < 16; i++)  Serial.write(payload[i] );
     Serial.println(" ");
    }
```

```
        char msgs[17];
        //SENT BLOCK
        for (byte i = 0; i < 16; i++) msgs[i] = (char)payload[i];
        Serial.print("Publish message: ");
        msgs[16] = '\n';
        Serial.write(msgs);
        client.publish("success/test1",  msgs);
        delay(1000);

  }else{
   Serial.println("Old Card ID Faild");
   }

 mfrc522.PICC_HaltA(); // Halt PICC
 mfrc522.PCD_StopCrypto1();  // Stop encryption on PCD
}

void reconnect() {
 while (!client.connected()) {
   Serial.print("Attempting MQTT connection...");
   String clientId = "ESP8266Client-";
   clientId += String(random(0xffff), HEX);
   if (client.connect(clientId.c_str())) {
     Serial.println("connected");

     client.subscribe("receive/test1");

   } else {
     Serial.print("failed, rc=");
     Serial.print(client.state());
     Serial.println(" try again in 5 seconds");
     delay(5000);
   }
 }
}

void setup() {
 Serial.begin(115200);       // Initialize serial communications with the PC
 SPI.begin();               // Init SPI bus
 mfrc522.PCD_Init();        // Init MFRC522 card
 Serial.println(F("Write personal data on a MIFARE PICC "));
 setup_wifi();
 client.setServer(mqtt_server, 1883);
 client.setCallback(callback);
}
```

```
void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  Serial.print(F(" PICC type: "));   // Dump PICC type
  MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
  Serial.println(mfrc522.PICC_GetTypeName(piccType));

  byte buffer[18];
  byte block;
  MFRC522::StatusCode status;
  byte len;

  //READ BLOCK

  block = 1;

  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1,
&key, &(mfrc522.uid)); //line 834
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

  status = mfrc522.MIFARE_Read(block, buffer, &len);
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
```

```
  }

  //PRINT BLOCK
  for (uint8_t i = 0; i < 16; i++) {
    Serial.write(buffer[i] );
  }

  Serial.println(" ");
  //SENT BLOCK
    for (byte i = 0; i < 16; i++) msg[i] = (char)buffer[i];
    Serial.print("Publish message: ");
    msg[16] = '\n';
    Serial.write(msg);
    client.publish("sent/test1", msg);

  Serial.println(" ");

  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();

}
```

ДОДАТОК Б

Вихідний код зчитує дані з RFID ключа та передає на сервер через WI-FI по протоколу MQTT.

```
import paho.mqtt.client as mqtt
import paho.mqtt.publish as publish
import pymysql
import random

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))

    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.
    client.subscribe("#")

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    topic = msg.topic
    payload = msg.payload.decode("utf-8", errors='ignore')
    print(topic)
    print(payload.split('\n')[0]+'\n')

    con = pymysql.connect(host='localhost',
                          user='ubuntu',
                          password='ubuntu',
                          database='rfidusers',
                          cursorclass=pymysql.cursors.DictCursor)


    type = topic.split('/')[0]
    device = topic.split('/')[1]

    if type=='sent':
        while True:
```

```
                newid = randomid()
                cur = con.cursor()
                cur.execute(f"SELECT * FROM users WHERE cardid='{newid}'")
                con.commit
                if cur.fetchone()==None:
                    break

            cur = con.cursor()
            sql = f"UPDATE users SET newcardid = '{newid}' WHERE cardid='{payload}'"
            cur.execute(sql)
            publish.single("receive/"+device, newid+payload, hostname="localhost")

        if type=='success':
            cur = con.cursor()
            sql = f"UPDATE users SET cardid = '{payload}' WHERE newcardid ='{payload}'"
            cur.execute(sql)

    def randomid():
        newid = ""
        for i in range(16):
            newid += chr(random.randrange(97, 97 + 26))
        return newid

    client = mqtt.Client()
    client.on_connect = on_connect
    client.on_message = on_message

    client.connect("localhost", 1883, 60)

    client.loop_forever()
```

ДОДАТОК В

HUAWEI
ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «ХУАВЕЙ УКРАЇНА»
LIMITED LIABILITY COMPANY "HUAWEI UKRAINE"
ТОВ «ХУАВЕЙ УКРАЇНА» / LLC "HUAWEI UKRAINE"
03124, Київ, вул. Радищева, буд. 10/14, (ідентифікаційний код: 33501267), тел. 498-77-18, факс 498-77-19
10/14, Radischeva street, Kiev, 03124, Ukraine, (Company Code: 33501267), tel. 498-77-18, fax 498-77-19

ЗАТВЕРДЖУЮ
Заступник директора з питань персоналу
ТОВ "Хуавей Україна"
_____ В.Б. Білавка
(підпис, ініціали, прізвище)

«___»_____2023 р.

Акт
про реалізацію результатів наукових досліджень
Данильченко Валентини Миколаївни

Матеріали роботи використано в науково-дослідних і дослідно-конструкторських роботах, що ведуться в ТОВ "Хуавей Україна", а саме, для вдосконалення персональної ідентифікації HUAWEI "Face Unlock".

В ТОВ "Хуавей Україна" використані наступні результати:

1. Впроваджено модель організації зберігання і обробки даних в МКПІ, що підвищує їх надійність на 23%.
2. Застосований метод і алгоритм підвищення надійності МКПІ з урахуванням особливостей апаратно-програмних платформ, що дозволило у 89% випадків корегувати дані в МКПІ.

Матеріали роботи «Методи обробки даних в мікрокомп'ютерних комплексах на основі персональної ідентифікації» дозволять вдосконалити модулі та алгоритми платформи для розробки мобільних додатків, яка надає інструментарій для вдосконалення персональної ідентифікації HUAWEI "Face Unlock, що є важливим внеском у розвиток

сучасних технологій та їх практичне впровадження в сферу мобільних додатків.

Заступник директора з питань
персоналу ТОВ "Хуавей Україна" В.Б. Білавка

ДОДАТОК Г

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної роботи ДУІКТ,
Доктор технічних наук, професор
_____Беркман Л.Н.
«21» ____вересня____2023 р.

АКТ

про реалізацію результатів дисертаційної роботи Данильченко Валентини Миколаївни на тему: «Методи обробки даних в мікрокомп'ютерних комплексах на основі персональної ідентифікації» у навчальному процесі Державного університету інформаційно-комунікаційних технологій

Комісія у складі:

Комісія у складі: голови комісії - директора Навчально-наукового інституту інформаційних технологій, доктора технічних наук, професора Бондарчука А.П., та членів комісії - завідувача кафедри Комп'ютерної інженерії, кандидата технічних наук, доцента Лащевської Н.О.;- завідувача кафедри Інженерії програмного забезпечення автоматизованих систем, доктора технічних наук, професора Сторчак К.П..; - завідувача кафедри Інженерії програмного забезпечення доктора технічних наук, доцента Замрій І.В. розглянула дисертаційну роботу Данильченко Валентини Миколаївни на тему: «Методи обробки даних в мікрокомп'ютерних комплексах на основі персональної ідентифікації» та публікації автора за матеріалами дисертаційної роботи. Результати впроваджено в начальний процес Навчально-наукового інституту Інформаційних технологій, а саме:

1. Ряд математичних теорем, які розширюють розуміння того, як можна використовувати завадостійкі коди в умовах обмежених ресурсів..

2. Методи порівняння, моделювання та математичне представлення змісту та структури досліджуваних об'єктів.

3. Модернізовані методи зберігання інформації та введення-виведення даних на основі нових методів кодування інформації, що дозволяє підвищити ефективність та надійність зберігання та обробки інформації.

4. Розроблені математичні моделі випадкового та послідовного вибору сторінок для обробки даних.

На основі аналізу представлених матеріалів комісія установила, що отримані наукові результати використовуються в навчальному процесі Державного університету інформаційно-комунікаційних технологій при викладанні навчальних дисциплін: «Моделювання IoT», «Теорія інформаційних процесів та систем».

Голова комісії

Директор ННІІТ
доктор технічних наук, професор А.П. Бондарчук

Члени комісії

Завідувач кафедри Комп'ютерної інженерії, кандидат технічних наук, доцент Лащевська Н.О.

Завідувач кафедри Інженерії програмного забезпечення автоматизованих систем, доктор технічних наук, професор Сторчак К.П.

Завідувач кафедри Інженерії програмного забезпечення, доктор технічних наук, доцент Замрій І.В.

Документ підписано у сервісі Вчасно (початок)
дисертація_Данильченко_В_М_.pdf

Документ підписано у сервісі Вчасно (продовження)
дисертація_Данильченко_В_М_.pdf

Документ відправлено: 09:25 26.12.2023

Власник документу

Електронний підпис
09:25 26.12.2023
Ідентифікаційний код: 3422101242
Данильченко Валентина Миколаївна
Власник ключа: Данильченко Валентина Миколаївна
Час перевірки КЕП/ЕЦП: 09:25 26.12.2023
Статус перевірки сертифікату: Сертифікат діє
Серійний номер: 382367105294AF9704000000F1C71F005332C801
Тип підпису: кваліфікований