Ємельянов В. О. Об’єктна модель програмного забезпечення захисту конфіденційних даних користувача / В. О. Ємельянов // Системи обробки інформації : зб. наук. праць. – Харків : ХУПС, 2012. – № 101. – С. 64-68.

УДК 004.42

В.О.Ємельянов

Севастопольський інститут банківської справи УАБС НБУ, Севастополь, Україна

ОБ’ЄКТНА МОДЕЛЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ КОНФІДЕНЦІЙНИХ ДАНИХ КОРИСТУВАЧА

Показана актуальність створення програмного забезпечення захисту конфіденційних даних (логін та пароль). Розроблена об’єктна модель програмного забезпечення захисту конфіденційної інформації за допомогою уніфікованої мови моделювання UML. Модель відображає основні абстракції предметної області, варіанти використання програмного забезпечення та організацію програмних модулів. Об’єктна модель є основою для побудови спеціалізованого програмного забезпечення.

Ключові слова: *об’єктна модель, програмне забезпечення, захист паролю, уніфікована мова моделювання (UML).*

Вступ та постановка завдання

Однією з важливіших проблем захисту та зберігання інформації є проблема витоку інформації, та персональних даних зокрема.

На теперішній час існує достатньо схожих програм, що розрізняються за ступенем захисту інформації, способом її зберігання та відображення (інтерфейс користувача). Призначення всіх цих програм допомогти користувачу надійно зберігати конфіденційні дані, наприклад, паролі для доступу в Інтернет, електронній пошті та іншим сервісам.

Використання однакового паролю для доступу до різних даних є небезпечним, бо виток пароля чи ключа при доступі до будь-яких даних, одночасно, дає доступ до всієї інформації.

Для вирішення цього завдання існують різні програмні засоби. Найбільш простим та популярним засобом із захисту конфіденційних даних є програмний продукт SCARABAY. Але у програмі відсутня можливість автоматичного заповнення форм та зберігання різних типів даних. Інші пакети для вирішення подібних завдань Password Commander і Password Boss мають засоби для автоматичного заповнення форм, обидві дозволяють створювати поля для введення особистих даних, але вони не є достатньо стійкими.

Виникає достатньо багатогранна проблема автоматизації зберігання паролів. Таким чином, існує необхідність розробки програмного забезпечення для захисту конфіденційних даних користувача.

Розробка об’єктної моделі програмного забезпечення

На сучасному етапі розвитку інформаційних систем і технологій структурі та організації програмного забезпечення приділяється особлива увага.

При розробці структури програмного забезпечення був використаний об'єктно-орієнтований підхід [1]. В основі об'єктно-орієнтованого підходу лежить об'єктна декомпозиція, тобто подання розроблюваного програмного забезпечення у вигляді сукупностей об'єктів, в процесі взаємодії яких, через передачу повідомлень, відбувається виконання необхідних функцій [2].

Специфікація розроблюваного програмного забезпечення об'єднує в собі наступні моделі:

1. Модель використання - являє собою опис функціональності програмного забезпечення з точки зору користувача.

2. Концептуальна модель - модель, що описує основні абстракції предметної області, які забезпечують необхідну функціональність ПЗ та їх взаємодія;

3. Модель реалізації - визначає реальну організацію програмних модулів і файлів.

Для побудови цих моделей був використаний уніфікована мова моделювання (UML).

У моделі використання проектована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою діаграми прецедентів.

Діаграма прецедентів розробленого програмного засобу наведена на рисунку 1. Актором виступає користувач.

Конфіденційні дані користувача будуть зберігатися у бінарному файлі (далі файл-контейнер). Для отримання доступу до інформації до файлу-контейнеру користувачу пропонується згенерувати ключі за допомогою введення паролю.

Згідно до вимог до програмного засобу роботу програми можна умовно розділити на роботу зі структурою даних і роботу з безпосередньо конфіденційними даними.

Взаємодія між користувачем та програмним засобом при введенні паролю наведена на діаграмі послідовностей, що наведена на рисунку 2.

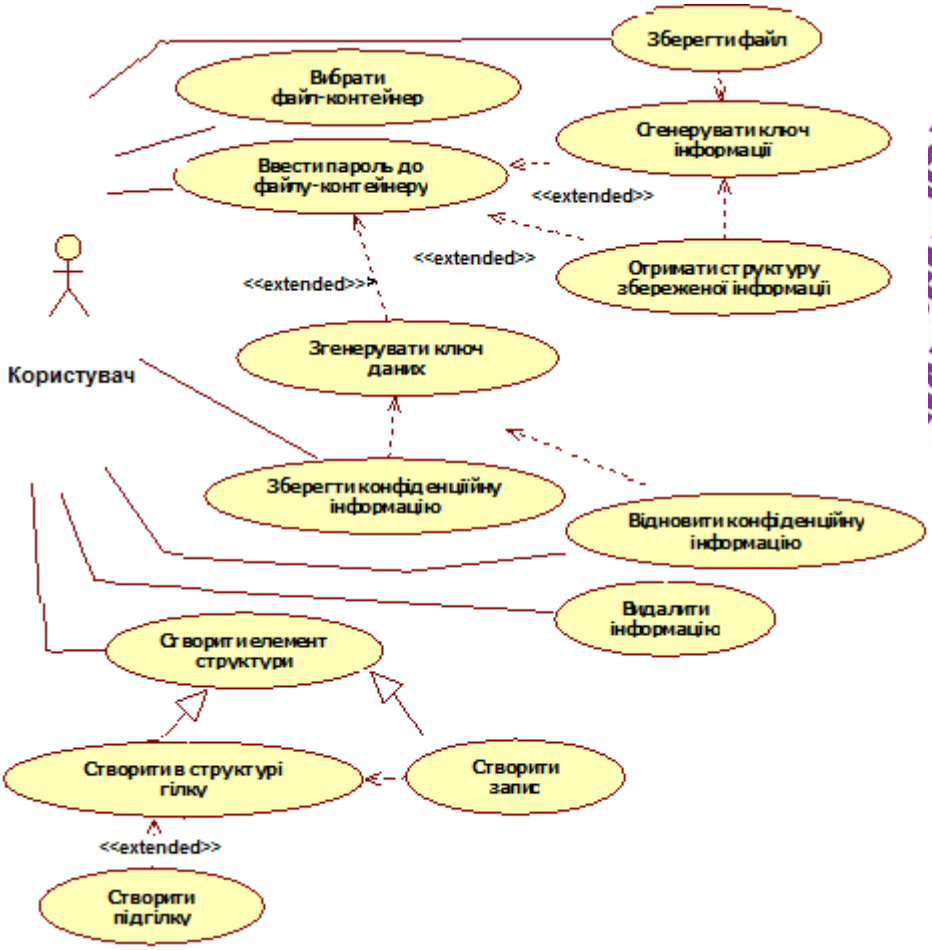

Рис. 1. Загальна діаграма прецедентів програмного засобу

Згідно до алгоритму генерації ключів допускається лише послідовне введення символів. У разі помилки під час вводу послідовності символів є можливість скидання до первинного стану. Редагування введеної послідовності неможливо. Користувач за допомогою головної форми програми викликає екранну клавіатуру. Форма екранної клавіатури відображається модально.

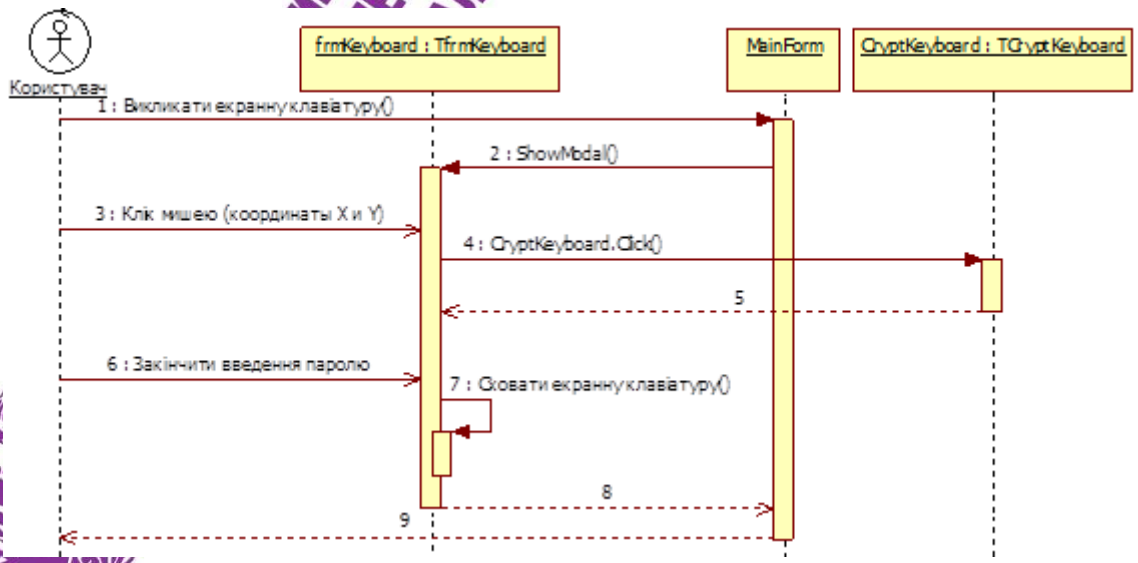

Рис. 2. Діаграма послідовності при введенні паролю

Користувач може ввести символ натисканням клавіші миші на формі екранної клавіатури, у цьому разі буде визвано метод TCryptKeyboard::Click().

Користувач може відмінити введення паролю або ж підтвердити його введення, у будь якому разі – це буде «завершення введення паролю». Вікно екранної клавіатури буде сховане до наступного виклику.

Згідно з обраною методикою шифрування AES-128 [3] та зберігання даних існує потреба зберігання в одному файлі декількох інформаційних потоків (інформаційний потік структури даних, безпосередньо потоки даних конфіденційної інформації).

Таким чином, необхідним є розробка алгоритму, що дав би змогу зберігати в файл та розрізнювати інформаційні потоки, зчитувати та видаляти дані.

Найпростішим рішенням є послідовне збереження даних у файл, кожен інформаційних потік буде додаватися в кінець файлу. Використання цього способу дасть змогу виконати усі поставлені задачі.

Таким чином, на підставі опису даної моделі представляється можливим побудувати концептуальну модель ПЗ. Концептуальна модель побудована на основі діаграм класів.

Робота з файлом-контейнером реалізована класом TStreamContainer. Діаграма класів представлена на рисунку 3.

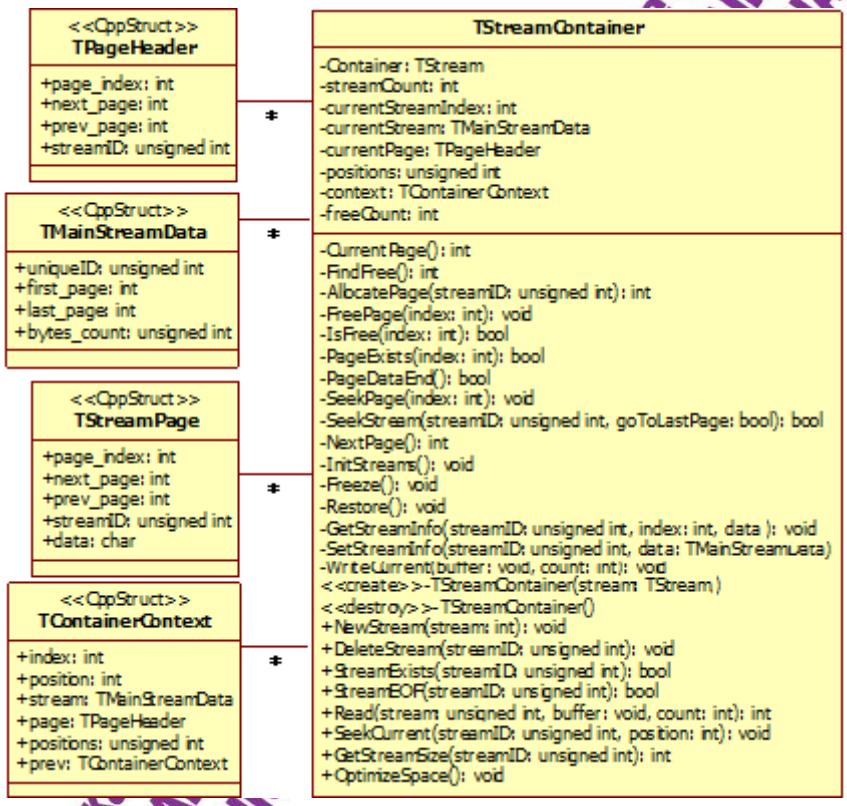

Рис. 3. Діаграма класів для TStreamContainer

Згідно з обраним алгоритмом шифрування AES-128 був реалізований модуль «crypt».

Модуль виконує шифрування даних з використанням створеного ключа (клас TAES128) та їх збереження до потоку (клас TStream). Окрім того модуль виконує шифрування даних, що збережені в потоці (клас TStream) з використанням створеного ключа (клас TAES128) та їх збереження до контейнеру (клас TStreamContainer) у інформаційний потік з завданим ідентифікатором.

Також призначенням модулю є зчитування даних, що збережені у інформаційному потоці з завданим ідентифікатором контейнера (клас TStreamContainer), їх дешифрування з використанням створеного ключа (клас TAES128) та збереження у в потоці (клас TStream).

Обчислення відбитка (128 біт) за алгоритмом MD5, блоку пам’яті заданої довжини також покладено на модуль crypt.

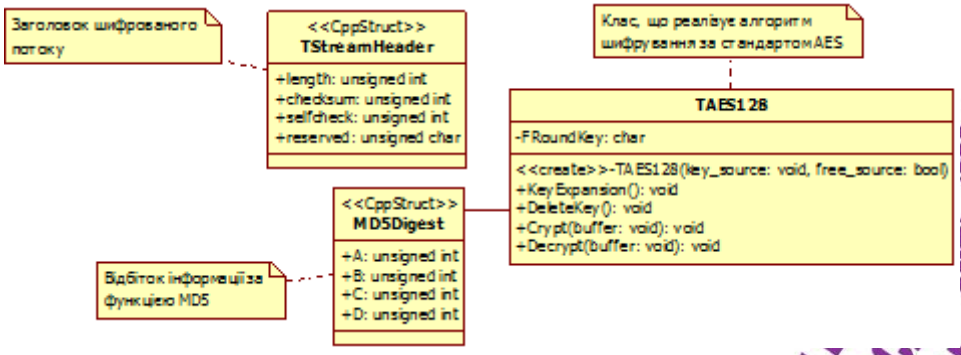

Рис. 4. Класи та структури модуля crypt

Створені раніше моделі відображали концептуальні аспекти побудови об'єктної моделі системи і відносилися до логічного рівня представлення. Основне призначення логічного представлення полягає в аналізі структурних і функціональних відносин між елементами моделі системи. Однак для створення конкретної фізичної системи необхідно деяким чином реалізувати всі елементи логічного представлення в конкретні матеріальні сутності. Для опису таких реальних сутностей призначений інший аспект модельного подання, а саме фізична подання об'єктної моделі. Для фізичного представлення побудована модель реалізації за допомогою діаграми компонентів, яка представлена на рисунку 5:

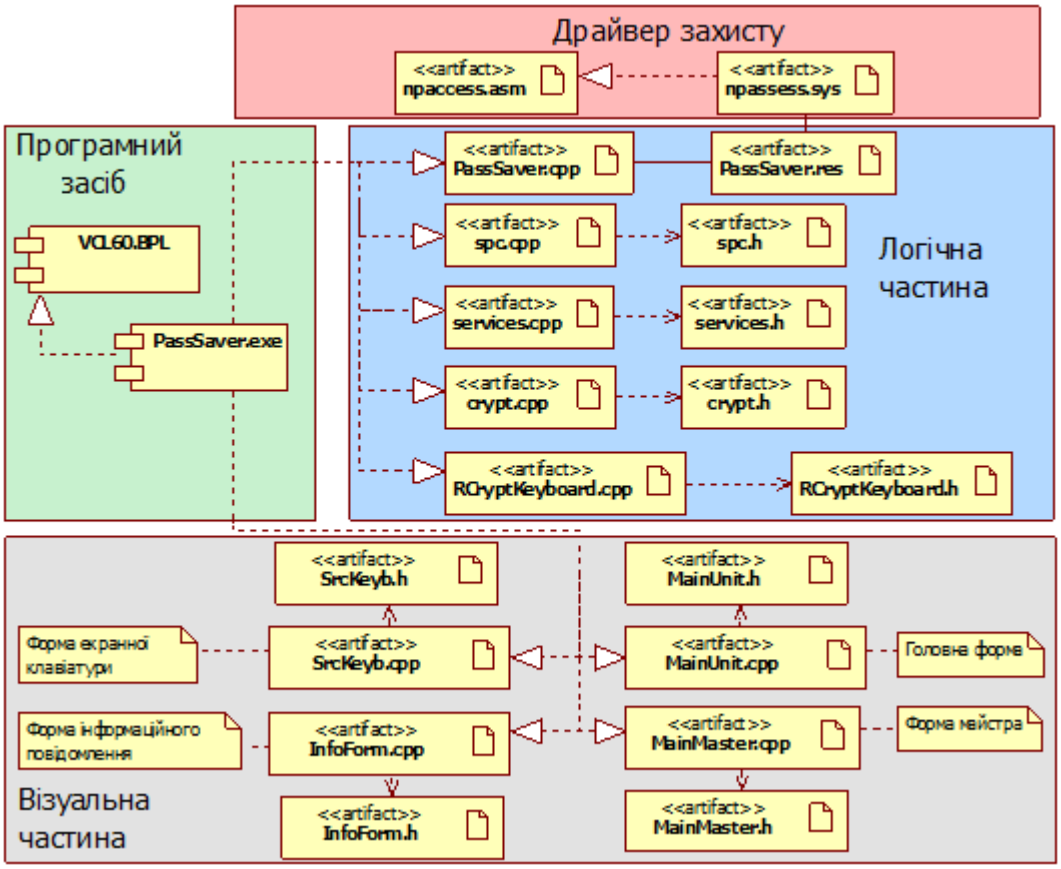

Рис. 5. Діаграма компонентів програмного забезпечення

Структуру умовно можна поділити на три частини:

– драйвер захисту;

– логічна частина (реалізує необхідні базові алгоритми);

– візуальна частина (забезпечує зв'язок користувача з логічною частиною).

Між собою частини зв'язані ієрархічно, візуальна частина використовує інтерфейси логічної, яка в свою чергу використовує драйвер захисту.

Захист програми від сканування пам’яті та графічного зображення головної форми реалізоване за допомогою драйверу режиму ядра Windows NT [4-5].

Принцип роботи захисту полягає в блокуванні роботи центрального процесору на момент відображення паролю. Під блокуванням роботи центрального процесору мається на увазі монопольне його використання драйвером на деякий, вибраний користувачем час.

Ініціалізація драйвера полягає в створенні віртуального пристрою. При завантаженні драйвера в пам'ять, викликається процедура DriverEntry, в якій необхідно розмістити відповідний код.

Якщо створення пристрою пройшло без помилок, то наступним нашим кроком буде створення символьного посилання, що вказує на пристрій. Для забезпечення безпеки користувач не може безпосередньо звертатися до пристрою, оскільки об'єкти ядра не доступні режиму користувача. Тому створюється символьне посилання, видиме з режиму користувача, поводження із запитом до якої викликає формування диспетчером завдань IRP-пакету і напрям його пристрою. Це дозволить коду режиму користувача дістати доступ пристрою.

У режимі ядра Windows не стежить за діями драйверів, і якщо драйвером був виділений будь-який ресурс – наприклад, блок оперативної пам'яті, – то тільки драйвер знає про цей блок пам'яті і при вивантаженні драйвера, інформація про нього втрачається і його вивантаження абсолютно унеможливлюється. Також в процедурі вивантаження драйвера передбачено видалення символьного посилання на пристрій, а також самого пристрою.

Драйвер розроблено за допомогою мови MacroAssembler та додатку KMDKit.

Висновки

Таким чином, побудована об'єктна модель програмного забезпечення захисту конфіденційних даних користувача відображає основні абстракції предметної області, варіанти використання програмного забезпечення, його фізичне уявлення, а також інформаційні потоки, що функціонують в програмному забезпеченні необхідному для процесу захисту таких даних користувача, як логін та пароль.

Список літератури

1. Грэхем И. Объектно-ориентированные методы. Принципы и практика. / И.Грэхем – М.: Вильямс, 2004. — 768 с.

2. Якобсон И. Унифицированный процесс разработки программного обеспечения. / И.Якобсон, Г.Буч, Дж Рамбо - СПб.: Питер, 2002. — 458 с.

3. Масленников М. Практическая криптография. / М.Масленников - СПб.: BHV, 2003. – 458 с.

4. Смит Г. Windows Driver Foundation: Разработка драйверов (пер. с англ. Таранушенко С.). / Г. Смит, П. Орвик - БХВ-Петербург, «Русская Редакция», 2008 – 880 с.

5. Шрайбер Свен Б. Недокументированные возможности Windows 2000 (+CD). / Свен Б Шрайбер.- СПб.: Питер, 2002 год – 544 с.

Рецензент:

Автори:

Ємельянов Віталій Олександрович *канд. техн. наук, асистент кафедри інформаційних технологій та систем СІБС УАБС НБУ, Севастополь, Україна, e-mail: V.yemelyanov@gmail.com*

ОБЪЕКТНАЯ МОДЕЛЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЗАЩИТЫ КОНФИДЕНЦИАЛЬНЫХ ДАННЫХ ПОЛЬЗОВАТЕЛЯ

В.А.Емельянов

Показана актуальность создания программного обеспечения защиты конфиденциальных данных (логин и пароль). Разработана объектная модель программного обеспечения защиты конфиденциальной информации с помощью унифицированного языка моделирования UML. Модель отражает основные абстракции предметной области, варианты использования программного обеспечения и организацию программных модулей. Объектная модель является основой для построения специализированного программного обеспечения.

Ключевые слова: *объектная модель, программное обеспечение, защита пароля, унифицированный язык моделирования (UML).*

THE OBJECT MODEL SOFTWARE OF THE PROTECTION OF CONFIDENTIAL USER DATA

V.A.Iemelianov

The urgency of creating software to protect confidential data (login and password). The object model of software to protect confidential information through a unified modeling language UML was designed. The model reflects the basic abstractions of the domain, the options of using the software and the organization of software modules. The object model is the basis for the construction of specialized software.

Keywords: *object model, software, password protection, unified modeling language (UML).*