Ємельянов В. О. Метод захисту конфіденційних даних користувача / В. О. Ємельянов // Системи управління, навігації та зв`язку : зб. наук. праць. – Київ : ДП"ЦНДІНУ", 2012. – № 1(21). – С. 186-190.

УДК 004.62

В.О.Ємельянов

Севастопольський інститут банківської справи УАБС НБУ, Севастополь, Україна

МЕТОД ЗАХИСТУ КОНФІДЕЦІЙНИХ ДАНИХ КОРИСТУВАЧА

Описана проблема захисту конфіденційної інформації (паролів та логінів). Обґрунтована актуальність розробки методу захисту паролів та логінів. Запропонована модифікована форма екранної клавіатури для захисту від «кейлоггерів». Описано алгоритм генерації ключів для розробленого програмного забезпечення. Запропоновано спосіб захисту інформації при збереженні даних на основі файлу-контейнеру. Описано механізм роботи з файл-контейнером.

***Ключові слова**: захист даних, генерація ключа, шифрування, файл-контейнер, екранна клавіатура.*

Вступ

Проблема захисту та зберігання персональних даних є актуальною проблемою сучасного суспільства.

Згідно з дослідженням [1], у ході якого були вивчені наслідки 31 витоку інформації, кожний витік будь-якої інформації у середньому наносить компанії збиток у розмірі 4,8 млн дол.

У всесвітній мережі Інтернет 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 або просто якусь секретну інформацію.

Використання однакового паролю для доступу до різних даних є небезпечним, бо витік пароля чи ключа при доступі до будь-яких даних, одночасно, дає доступ до всієї інформації. Є два можливі рішення цієї проблеми – використовувати різні методи генерації ключа з одного паролю, або використовувати різні паролі. Перший метод потребує великих витрат на створення достатньо крипостійких алгоритмів, а другий метод потребує зусиль від користувача, а саме він полягає в запам'ятовуванні великої кількості інформації.

Для вирішення цих завдань 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. З одного боку – програмне рішення, що реалізує цю функцію, повинно надійно зберігати конфіденційні дані (повинен використовуватися крипостійкий алгоритм), також повинно бути захищена від витоку інформації в процесі роботи (під час її вводу, виводу), з іншого – бути мобільним, щоб користувач міг у будь який момент скористатися програмою.

Постановка завдання

Розглянемо методи щодо захисту конфіденційних даних користувача.

По-перше, необхідно зробити захист від викрадання інформації з пам'яті. Це можна реалізувати двома методами.

Перший метод полягає в захисті області пам'яті від читання іншими процесами через драйвер режиму ядра Windows NT. Проте навіть такий метод не може забезпечити стовідсоткову надійність. Вже давно розроблені такі налагоджувані (наприклад, SoftICE), що може в будь-який момент за мережною командою з головної машини зупинити роботу веденої системи і вилучити будь-які дані.

Другий метод – це не зберігати в пам'яті дані в тому виді, в якому їх легко витягти з програми. Враховуючи що передбачається зберігати тільки текстову інформацію актуально безпосередньо при уведенні її користувачем перетворювати в графічну, причому алгоритм перетворення повинен забезпечувати захист від автоматичного зворотного перетворення і її однозначність для людського сприйняття. Проте цей метод може виявитись дуже вразливим місцем програми, якщо не передбачити захисту від автоматичного сканування графічної інформації з робочої області програми.

Таким чином, є необхідність розробки методу для захисту конфіденційних даних користувача.

Розробка методу захисту від несанкціонованого витоку даних

При розробці методу потрібно підвищувати криптостійкість шифру, а також запропонувати можливість захисту від програм «кейлогерів».

Для підвищення криптостійкості шифру в методі додається елемент випадковості в графічну інформацію. Якщо зловмисник буде знати частину

розшифрованої конфіденційної інформації користувача, це не дасть йому можливості скоротити підбір ключа.

Для захисту програми від так званих «кейлогерів», програм, що відстежують події натискання клавіш на клавіатурі організується екранна клавіатура і забезпечується введення за допомогою миші.

Структурна схема екранної клавіатури подана на рисунку 1.

Розташовування випадковим шляхом кнопок на формі екранної клавіатури з метою унеможливити відстеження паролю за положенням курсору відносно вікна дуже утруднить уведення.

Має сенс розробити круглу екранну клавіатуру, радіально розташувати кнопки й забезпечити їхнє переміщення під час уведення. Початковий кут повороту буде вибиратися випадковим шляхом. Переміщення повинне бути не занадто швидким інакше це погіршить ергономіку програми. Структурна схема модифікованої форми екранної клавіатури представлена на рисунку 1.

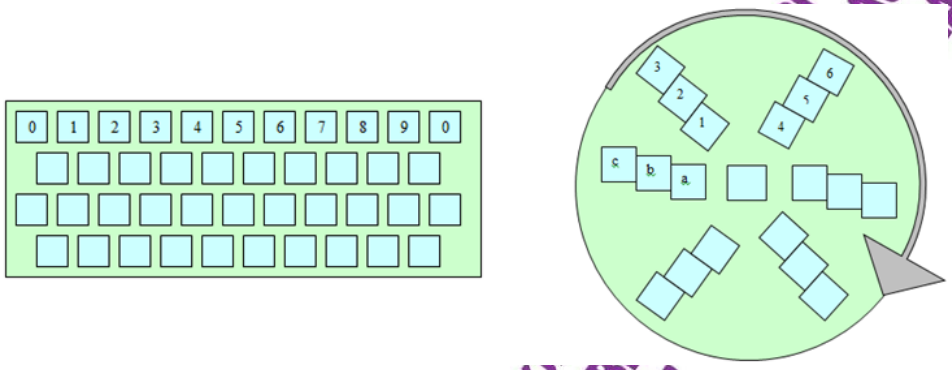

Рис. 1. Структурні схеми форми «Екранна клавіатура» та модифікованої форми «Екранна клавіатура»

Також треба виключити зберігання назви кнопок як текст у властивостях вікна Windows. Зберігати назви кнопок необхідно як зображення.

Згідно до алгоритму генерації ключів допускається лише послідовне введення символів паролю. У разі помилки під час вводу послідовності символів є можливість скидання до первинного стану. Редагування введеної послідовності неможливо згідно до розробленого алгоритму генерації ключів.

Відповідно до вибраного алгоритму шифрування AES-128, що потребує 128-бітний ключ, вибрємо алгоритм MD5 з довжиною 128 біт [2-3].

На рисунку 2 представлено запропонований алгоритм генерації ключа інформації.

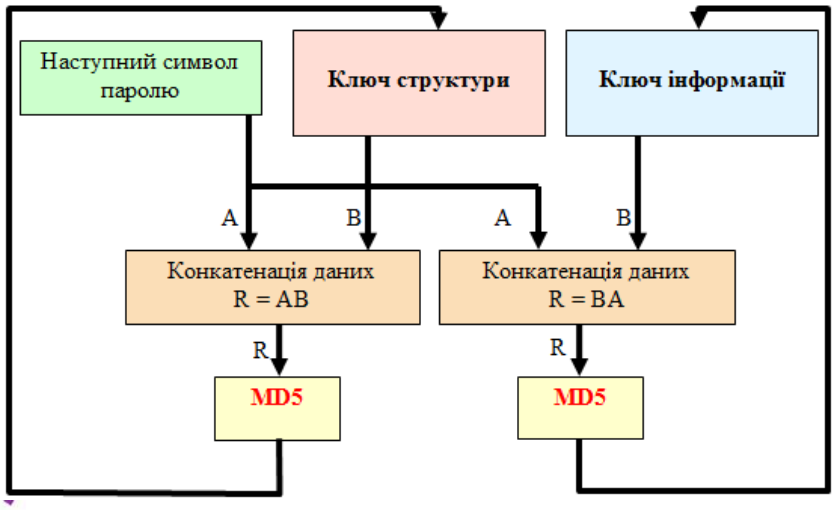

Рис. 2. Структурна схема алгоритму генерації ключів

Відповідно до представленої структурної схеми, генерація ключа відбувається нерозривно з введенням паролю.

Ключі структури та інформації ініціалізуються деяким початковим значенням.

Відповідно до вимог захисту від модифікації програми, це значення є відбитком виконуваного файлу програми за алгоритмом MD5. Якщо опція захисту від модифікації відключена, ключі структури та інформації ініціалізуються нулями.

Після кожного введення символу проводиться конкатенація даних. Для генерації ключа структури введений символ додається перед ключем, для ключа інформації – після ключа. В обох випадках є пос-

лідовність з 16 + 1 = 17 байт, для яких вираховується відбиток за алгоритмом MD5 [4]. Таким чином отримуємо наступні ключі – інформації та структури. Зворотна генерація неможлива (тобто неможливо знаючи введений символ отримати значення ключа до введення символу).

Реалізація захисту інформації при збереженні даних

Згідно з обраною методикою шифрування та зберігання даних є потреба зберігання в одному файлі декількох інформаційних потоків (інформаційний потік структури даних, безпосередньо потоки даних конфіденційної інформації).

Тому 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 вкінець файлу. Використання цього способу дасть змогу виконати усі поставлені задачі. Проте видалення інформаційного потоку залишає вільне місце у файлі, це буде збільшувати розмір кожен раз при внесенні нового інформаційного потоку.

Виникає проблема локалізації вільного простору та оптимізація розміру файлу.

Вирішенням цієї проблеми є створення нового файлу, дані в якому будуть копією даних збережених у попередньому, але розташованих в оптимізованому порядку.

Цей метод є простим для реалізації, проте виникає потреба обробки усіх даних, збережених у файлі, у тому числі і тих, позиція яких є оптимальною з точки зору розміру файлу.

Оптимальним з точки зору мінімізації оброблюємих даних є заповнення вільного простору інформацією з іншого потоку, тобто виникає необхідність сегментування інформаційних потоків.

При такому підході виникає необхідність розрізнення інформаційних потоків. Одним з підходів є створення таблиці розміщення даних та відповідності даних потоку.

Доцільним є стандартизація збереження інформації про розміщення сегментів потоку, тобто дані про розміщення необхідно зберігати так само як і звичайні інформаційні дані.

Отже, оберемо посторінковий метод зберігання даних [5]. Дані у файлі будуть представлені як відображено на рисунку 3.

Рис. 3. Структура розміщення даних у файлі-контейнері.

Згідно з обраним алгоритмом шифрування даних, що потребує дані, розбиті на блоки по 128 біт (16 байт), є доцільним обрати розмір сторінки кратним 16 байтам. Таким чином буде виключено ситуацію, коли дані одного блоку буде розташовано на різних сторінках, що дасть приріст продуктивності обробки даних.

Для зв'язку окремих сторінок у інформаційні потоки оберемо зв'язок шляхом маркування кожної сторінки. Виходячи з того, що дані на кожній сторінці будуть розбиті на блоки по 16 байт, то розмір маркеру також має бути кратним 16 байт. Структура маркеру сторінки представлена на рисунку 4.

Рис. 4. Структура сторінки: а – структура маркеру, б – загальна структура сторінки

Організація зв'язку сторінок у інформаційні потоки реалізована шляхом двозв'язного списку – це дасть змогу пришвидшити позиціонування в інформаційному потоці.

Абсолютний індекс сторінки – позиція розміщення сторінки у файлі, поділена на розмір сторінки.

Відносний індекс сторінки – відповідає номеру сторінки в інформаційному потоці. Перша сторінка має індекс 0. Кожна наступна – індекс попередньої, інкрементований на одиницю.

Якщо абсолютний індекс наступної сторінки дорівнює –1, поточна сторінка є останньою в інформаційному потоці.

Якщо абсолютний індекс попередньої сторінки дорівнює –1, поточна сторінка є першою в інформаційному потоці.

Ідентифікатор потоку – унікальний індекс інформаційного потоку. В маркері використовується

для перевірки цілісності даних. Значення цього поля повинні бути однакові у межах усіх сторінок інформаційного потоку.

На рисунку 5б наведена загальна структура сторінки. Маркер повинен бути розташований перед даними. Це є доцільним з точки зору оптимізації позиціонування. Якщо зчитано маркер, і є потреба зчитати дані, то в такому разі повторне позиціонування не потрібне – виконується послідовне зчитування даних. В іншому випадку (коли маркер буде розташовано опісля даних) виникне потреба повторного позиціонування на початок сторінки.

Проте лише зв'язки сторінок у списки недостатньо для повнофункціональної реалізації збереження інформації. Існує необхідність зберігання даних щодо потоків, а саме кількість байт (що необов'язково є кратною кількості даних на сторінці), початкової та кінцевої сторінки (для швидкого позиціонування) та ідентифікатора потоку. Структура даних, що має бути асоційована з інформаційним потоком приведена на рисунку 5.

Довжина структури даних, наведених на рисунку 6 складає 16 байт, тобто співпадає з розміром блоку даних шифрування – розмір сторінки є кратним довжині структури. Це є оптимальним з точки зору обробки даних – запис такої структури гарантовано буде розташовано на одній сторінці.

Рис. 5. Структура даних, асоційованих з інформаційним потоком

Виходячи з прийнятого принципу стандартизації роботи з даними файлу-контейнеру для збереження інформації про потоки виникає необхідність розділення інформаційних потоків на головний та підрядні.

Нульова сторінка файлу-контейнеру резервується для головного потоку. Головний потік містить як мінімум один запис про самого себе.

При створенні нового файлу-контейнеру, він має містити одну сторінку. Дані наведені в таблиці 1.

Таблиця 1

Початкові дані для створення нового файл-контейнеру

Маркер	
Відносний індекс	0
Абсолютний індекс попередньої сторінки	–1
Абсолютний індекс наступної сторінки	–1
Ідентифікатор потоку	0x80000000
Дані (згідно зі структурою, поданою на рис. 6)	
Ідентифікатор потоку	0x80000000
Початкова сторінка	0
Кінцева сторінка	0
Кількість байт	16

Для головного інформаційного потоку зарезервовано ідентифікатор SYSSTREAM_MAIN (числове значення 0x80000000).

Кожна сторінка, окрім нульової може бути звільнена зміною відносного індексу на –1. Звільнені сторінки можуть бути заново виділені іншим інформаційним потокам. Якщо вільних сторінок не знайдено, створюється нова сторінка вкінці файлу-контейнеру.

Такий підхід дає змогу реалізувати оптимізацію розміщення інформаційних потоків та вільного простору, а саме їх дефрагментацію.

Дефрагментований вільний простір переміщений вкінець файлу контейнеру може бути безпечно видалений.

Висновки

Таким чином, розроблено метод, який здійснює забезпечення інформаційної безпеки конфіденційних даних, таких як ім'я користувача і його пароль.

Алгоритм генерації ключів за паролем, що є складовим методу введення інформації за допомогою радіальної екранної клавіатури, розроблений так, що дає змогу не зберігати пароль протягом його введення до системи, що унеможливлює виток його стороннім особам.

Список літератури

1. 2006 Annual Study: Cost of a Data Breach [Електронний ресурс]: Режим доступу – http://www.michiganbusiness.us/showcompany.php?id=38903

2. Фергюсон Н. Практическая криптография. / Н. Фергюсон, Б. Шнайер - К.: Диалектика, 2004 – 432 с.
3. Масленников М. Практическая криптография. / М.Масленников - СПб.: BHV, 2003. – 458 с.
4. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. / О. Н.Василенко – М.: МЦНМО, 2006. – 336 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*

МЕТОД ЗАЩИТЫ КОНФИДЕНЦИАЛЬНЫХ ДАННЫХ ПОЛЬЗОВАТЕЛЯ

В.А.Емельянов

Описана проблема защиты конфиденциальной информации (паролей и логинов). Обоснована актуальность разработки метода защиты паролей и логинов. Предложена модифицированная форма экранной клавиатуры для защиты от «кейлоггеров». Описан алгоритм генерации ключей для разработанного программного обеспечения. Предложен способ защиты информации при хранении данных на основе файла-контейнера. Описан механизм работы с файл-контейнером.

Ключевые слова*: защита данных, генерация ключа, шифрование, файл-контейнер, экранная клавиатура.*

METHOD FOR OF THE CONFIDENTIAL USER DATA PROTECTION

V.A.Iemelianov

The problem of protecting confidential information (passwords and logins) was described. The urgency of developing a method for protecting passwords and logins was described. The modified form of on-screen keyboard to protect against "keylogging" was proposed. An algorithm for generating keys for the software developed. A method for protecting data in storage-based data container file. The mechanism of working with a file container was described.

Keywords*: data protection, key generation, encryption, file container, on-screen keyboard*