УДК 681.3.07

О. М. Хошаба
В. В. Войтко
А. С. Штокал

РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ДЛЯ УПРАВЛІННЯ ПАРОЛЯМИ ТА ЇХ ШИФРУВАННЯ

Вінницький національний технічний університет

Анотація

Розглянуто основні особливості розробки веб-додатку для шифрування та збереження паролів. Описано головний функціонал роботи додатку, подано модель бази даних та діаграму процесу додавання паролів. Розглянуто методи шифрування паролів на прикладі алгоритмів SHA512 та HMAC.

Ключові слова: *веб-додаток, шифрування, пароль, шифрування паролів*.

Abstract

The article discusses the main features of developing a web application for encrypting and storing passwords. The main functionality of the application is described, the database model and the diagram of the process of adding passwords are given. Methods of password encryption on the example of SHA512 and HMAC algorithms are considered.

Keywords: *web-application, encryption, password, password encryption*.

Вступ

Щоб активно користуватися послугами в мережі Інтернет у більшості випадків потрібен акаунт на певному веб-сайті, але через їх велику кількість стає дедалі складніше запам’ятати дані для входу до всіх власних акаунтів. Крім того, щоб захиститися від зламів хакерів, потрібно використовувати складні паролі і унікальні дані для різних веб-сайтів, що породжує проблему необхідності запам’ятовування чи зберігання всіх паролів доступу. Тому актуальним є питання розробки і використання програмного забезпечення, призначеного для зручного зберігання пароля. Сьогодні наявними є багато програмам подібного призначення, проте їх функціонал є досить обмеженим чи недоступним у безкоштовній версії.

Технологія шифрування стає доцільною, оскільки існує ймовірність зламу веб-сервісів, а користувач може стати жертвою фішингу. Компанії повинні хешувати пароль кожного разу, коли користувач його водить, проте не всі ресурси забезпечують таку можливість. Деякі компанії не приділяють належної уваги хешуванню, що провокує небезпеку зламу хакерами. Важливим також стає процес генерування складних паролів, що містять набір великих та малих літер, цифр і символів.

Запропонований додаток забезпечує користувачеві легкий доступ до своїх паролів і високий рівень захисту від несанкціонованого доступу.

Розробка веб-додатку для управління паролями і їх шифрування

Серед головних критеріїв аналізу функціоналу розроблюваного веб-ресурсу виокремимо наступні: можливість 256-бітового AES-шифрування, наявність протоколів з нульовим розголошенням, забезпечення аутентифікації. Наявні аналоги мають низку недоліків: блокування акаунту в цілях безпеки, блокування ір адреси, щоб запобігти спробам зламу [1].

Деякі з наявних програмних ресурсів не передбачають використання мастер-пароля. Запропоноване рішення передбачає використання мастер-пароля до шифрування всіх інших паролів, що підвищує безпеку такого додатку. Кожна наступна зміна мастер-пароля обумовлює перешифрування всіх паролів. Крім того, не всі з наявних програм управління паролями забезпечують можливість спільного доступу та експорт паролів до файлу.

Першим етапом проєктування веб-додатку стало визначення його функціоналу, а саме:

- вибір способу шифрування під час реєстрації нового користувача: SHA512 або HMAC;

- зміну пароля для входу в програму;
- функціонал додавання паролів користувачів;
- функціонал перегляду збережених паролів;
- розшифрування пароля, якщо користувач просить його відобразити;
- реєстрація спроб логування користувачами (запис таких даних, як час входу, результат входу (успішно, не вдалося), IP-адреса користувача);
- подання користувачеві даних про останній успішний і невдалих вхід;
- збереження інформації про кількість наступних неправильних пробних входів користувачів;
- у випадку дворазової невдалої спроби входу користувача час очікування на наступну спробу збільшується до 5 секунд;
- у випадку триразової невдалої спроби входу користувача час очікування на наступну спробу збільшується до 10 секунд;
- у випадку, якщо користувач не зміг увійти в свій акаунт чотири рази, обліковий запис користувача блокується на 2 хвилини;
- у випадку успішного входу в систему число неправильних входів скидається до нуля;
- перевірка IP-адреси, якою користується користувач – підрахунок правильних та неправильних спроб входу;
- у випадку, якщо користувач не зміг увійти щонайменше чотири рази з однієї і тої самої адреси – IP адреса блокується назавжди;
- у випадку успішного входу з конкретної IP-адреси, кількість наступних неправильних входів з цієї адреси скидається до нуля;
- можливість зняття блокування IP-адреси через запит до адміністраторів.

Усі процеси, що виконуються веб-сайтом, зручно описати за допомогою діаграмами Bussiness Process Model and Notation (BPMN). Вони однозначні, зрозумілі та гнучкі, що полегшує розуміння функціоналу програми. Рисунок 1 ілюструє процес додавання паролів.

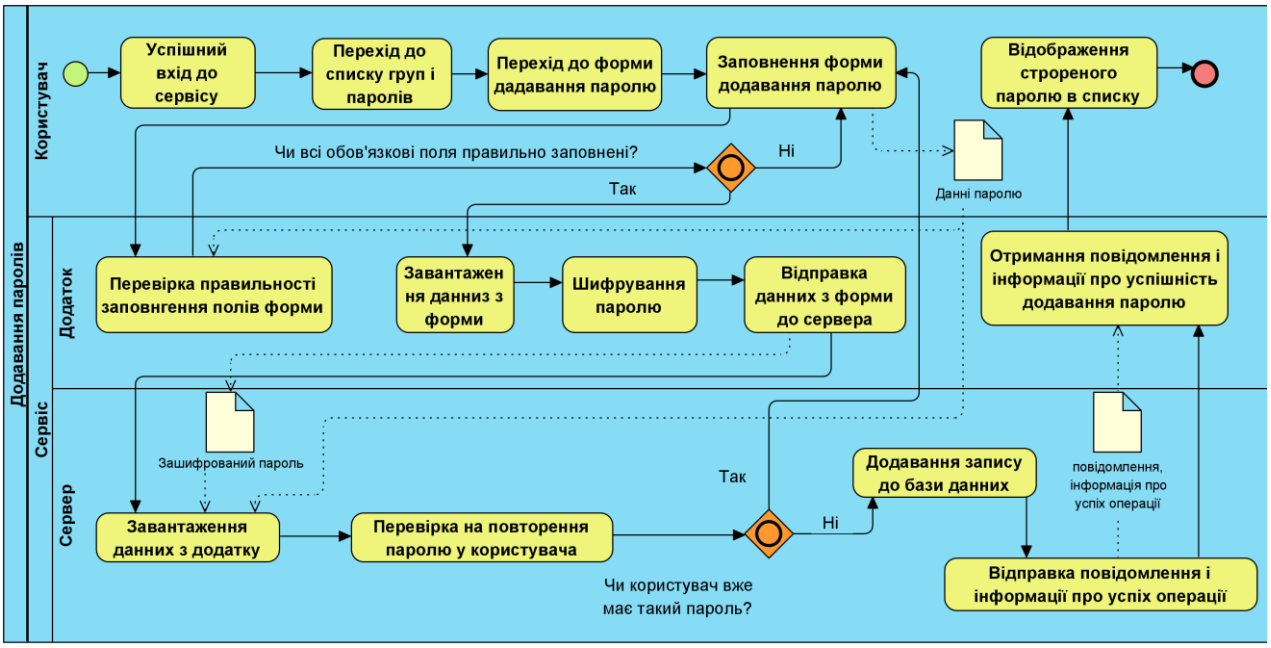

Рис. 1. Діаграма процесу додавання паролів

Важливим етапом у розробці програми є визначення нефункціональних вимог, які не стосуються функціоналу веб-додатку, але визначають спосіб його роботи:

- веб-сайт повинен підключатися до бази даних SQL;
- для користування веб-сайтом потрібне з'єднання з Інтернетом;
- веб-додаток є доступним у браузері;
- паролі в базі даних зберігаються у зашифрованому вигляді;
- окремо існують пароль для входу та паролі доступу для кожного рівня безпеки;
- ці паролі зберігаються у хеш-формі і є ключами шифрування;

- веб-додаток використовує алгоритм SHA-512 як хеш-функцію;
- веб-додаток використовує алгоритм AES-256 для шифрування паролів.

Програмний додаток побудований на реляційній базі даних, схема якої подана на рисунку 2.

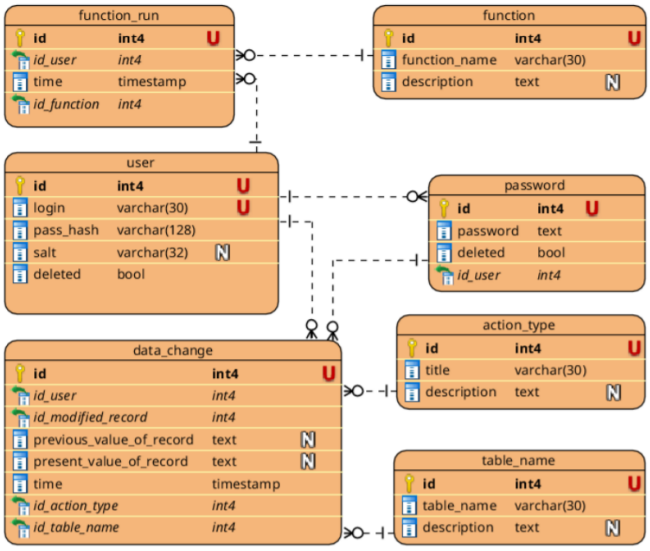

Рис. 2 Схема структури бази даних

Оскільки веб-додаток має мікросервісну архітектуру, то варто окремо розглянути частини бекенд та фронтенд. Для бекенду основним фреймворком було обрано Spring Boot [2]. Використання Spring Boot спрощує й оптимізує програмну розробку у порівнянні з використанням Java. Для розробки фронтенду основним фреймворком було обрано React, який є простим в освоєнні, має зрозумілий та лаконічний синтаксис.

Алгоритми шифрування

У додатку було використано два основні алгоритми шифрування. SHA512 – це хеш-функція, яка обчислює хеш-значення для набору даних [3]. Хеш обчислюється за спеціальним алгоритмом без додаткових даних. Це робить можливими деякі атаки, особливо при використанні таблиць. Тому під час перемішування додають сіль. Сіль – це послідовність байтів/символів, які додаються до паролю перед хешуванням. Важливо змінювати сіль під час зміни пароля.

Сіль зберігається в базі даних. Для кращого захисту паролів можна додати перець (випадкову послідовність байтів/символів, яка зберігається всередині програми, а не в базі даних). Можна зберігати перець безпосередньо у коді програми або у файлі конфігурації.

HMAC – це код автентифікації повідомлень на основі хешу [4]. Він обчислює хеш даних, але з використанням секретного ключа. Це означає, що тільки власник ключа може обчислити хеш. HMAC має низку переваг, зокрема велику стійкість до атак. До переваг алгоритму HMAC також можна віднести зручність його використання для високопродуктивних систем, зокрема маршрутизаторів, завдяки використанню хеш-функцій, які швидко обчислюються та перевіряються, на відміну від систем із відкритим ключем. Цифрові підписи є більшими за HMAC, проте HMAC забезпечують порівняно вищий рівень безпеки. HMAC використовуються в адміністраціях, де системи відкритих ключів заборонені [5]. До недоліків HMAC відносять використання спільного ключа. Якщо ключ відправника або одержувача пошкоджено, зловмисникам буде легко створювати несанкціоновані повідомлення.

Висновки

Розроблюваний веб-додаток для управління паролями і їх шифрування призначений для підвищення рівня безпеки збереження паролів користувачів. Функціонал веб-додатку дозволяє додавати нові паролі, використовуючи методи шифрування HMAC і SHA512. У додатку передбачена можливість змінювати головний пароль та ділитися паролями з іншими користувачами, блокувати окремого користувача та ip-адреси після n разового невдалого логування. Паролі зберігаються на декількох рівнях безпеки, тому навіть у випадку, коли неавторизована особа отримає доступ до

облікового запису, це не забезпечує можливості прочитати всі паролі. Використання найбезпечніших алгоритмів шифрування дозволяє забезпечити високий рівень захисту паролів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Dick K., XML: A Manager's Guide, Addison-Wesley Professional, Second Edition, 2002.
2. Iliescu O., Pro Java ME Apps: Building Commercial Quality Java ME Apps, Apress, 2011.
3. Abdalla, M., & Pointcheval, D. (2005). Simple Password-Based Encrypted Key Exchange Protocols. Topics in Cryptology – CT-RSA 2005, 191–208.
4. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, May 1992, pp. 72–84. IEEE Computer Society Press, Los Alamitos (1992).
5. Kobara, K., Imai, H.: Pretty-simple password-authenticated key-exchange under standard assumptions. IEICE Transactions E85-A(10), 2229–2237 (October 2002).

Хошаба Олександр Мирославович – кандидат технічних наук, доцент, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, e-mail: Oleksandr.Khoshaba@gmail.com.

Войтко Вікторія Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: dekanfki@i.ua.

Штокал Алла Сергіївна – студентка групи 1ПІ-19м, факультет інформаційних технологій і комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: alla.shtokal@gmail.com.

Oleksandr Khoshaba – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksandr.khoshaba@vntu.edu.ua.

Viktoriia Voitko – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: dekanfki@i.ua.

Alla Shtokal – student of 1ПІ-19m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: 1pi15b.shtokal@gmail.com.