DOI: https://doi.org/10.33216/2222-3428-2023-24-1
УДК: 004.056.5

Деркач М.В., Мишко О.Є.

ВИКОРИСТАННЯ АЛГОРИТМУ ШИФРУВАННЯ AES-256-CBC ДЛЯ ЗБЕРІГАННЯ ДАНИХ АВТЕНТИФІКАЦІЇ АВТОНОМНОГО ПОМІЧНИКА

У статті розглянуте актуальне питання розробки автономного помічника зберігання даних автентифікації, що допоможе забезпечити безпеку та надійність доступу до конфіденційної інформації та запобігти можливим крадіжкам даних. Оскільки у зв'язку зі зростанням використання онлайн-сервісів та електронної комерції, ризик несанкціонованого доступу до особистих даних користувачів значно збільшується. Для реалізації розробки використано мову програмування PHP, фреймворк Lumen, Telegram Bot API, Linux, Git/GitHub, в якості СУБД використано MySQL. Розглянуто протокол MTProto, який використовується месенджером Telegram, а також методи шифрування, які застосовуються у фреймворку Lumen, а саме методи шифрування та дешифрування даних на основі майстер-ключа, що дозволило забезпечити безпеку передачі та збереження даних автентифікації. Всі зашифровані значення шифруються за допомогою OpenSSL і шифру AES-256-CBC, додатково всі зашифровані значення підписуються кодом автентифікації повідомлення (MAC), щоб виявити будь-які зміни в зашифрованому рядку. Розроблений автономний помічник зберігання даних автентифікації має такі функціональні можливості: автентифікація користувача, управління паролями, швидкий пошук, імпорт та експорт даних, захист даних. А також має важливу перевагу перед існуючими засобами зберігання даних, такими як зберігання на локальному пристрої, використання хмарних сервісів, використання менеджерів паролів, оскільки забезпечує високий рівень захисту конфіденційної інформації користувачів за допомогою сучасних технологій шифрування та зменшення ризику витоку персональних даних при використанні онлайн-сервісів. Після реалізації системи, було проведено тестування та продемонстрована взаємодія автономного помічника з користувачем.

Ключові слова: *автономний помічник, алгоритм шифрування, безпека, збереження даних автентифікації, чат-бот.*

Вступ. Одним із розповсюджених напрямів розвитку IT-галузі є автономні помічники зберігання даних автентифікації, які забезпечують безпеку та уникнення крадіжки інформації. Автономний помічник зберігання даних автентифікації - це програмне забезпечення, яке зберігає дані користувача, такі як ім'я користувача та пароль, і забезпечує їх захист від несанкціонованого доступу. Це робиться шляхом шифрування даних та застосування різних алгоритмів безпеки. Одним із варіантів використання автономного помічника зберігання даних автентифікації є його інтеграція з чат-ботами. Чат-боти - це автоматизовані програми, які можуть виконувати різноманітні завдання, наприклад, надсилати повідомлення користувачам, відповідати на запитання та виконувати інші дії. Підтримку чат-ботів мають усі популярні месенджери, такі як Telegram, Facebook Messenger, WhatsApp, Viber та інші. Чат-боти дозволяють користувачам отримувати швидкий та ефективний доступ до інформації. Розробка автономного помічника для збереження даних автентифікації допоможе забезпечити безпеку та зручність управління паролями, сприяти ефективному пошуку імовірних рішень, а також створити зручний інструмент для керування даними автентифікації.

Аналіз останніх досліджень і публікацій. У даний час, коли більшість сервісів вимагають від користувачів реєстрації з використанням електронної пошти та пароля, дуже важливо зберігати ці дані в надійному та захищеному місці. Для зберігання даних автентифікації можуть використовуватись різноманітні методи:

1. Зберігання на локальному пристрої.
2. Використання хмарних сервісів [1,2].
3. Використання менеджерів паролів [3,4].

Кожен з цих методів має свої переваги та недоліки, однак головні ризики та загрози пов'язані з безпекою конфіденційних даних, а саме:

- Зловмисник може отримати паролі, обходячи шифрування та брандмауери.
- Атаки з використанням фішингу.
- Злам паролів.

Для захисту від цих загроз можуть бути застосовані наступні методи:

- Використання складних паролів.
- Використання двофакторної автентифікації.
- Використання VPN.

Автономний помічник на основі чат-боту для збереження даних автентифікації у месенджерах є кращим варіантом, оскільки це дозволить забезпечити уніфікований та зрозумілий інтерфейс для користувачів, спрощуючи процес спілкування та взаємодії з помічником, а головне може зберігати паролі та інші дані в зашифрованому вигляді, що дозволяє уникнути їхнього витоку чи викрадення. Крім того, такий автономний помічник може допомогти згенерувати складні паролі, які важко підібрати зловмисникам.

Мета статті. Розробити автономний помічник для забезпечення безпеки передачі та збереження даних автентифікації на основі фреймворк Lumen, який підтримує методи шифрування OpenSSL та AES-256-CBC.

Основний зміст роботи. Автономний помічник для збереження даних автентифікації має такі функціональні можливості:

- *Автентифікація користувача*. Забезпечення безпечного ідентифікування та автентифікації користувачів, включаючи можливість створення основного паролю.
- *Управління паролями*. Можливість створення, редагування, перегляду та видалення паролів для різних акаунтів і служб. Помічник може зберігати паролі в безпечному форматі та надає зручний інтерфейс для їх управління.
- *Швидкий пошук*. Функціонал пошуку, що дозволяє швидко знаходити потрібні дані автентифікації за назвою. Це допоможе зручно організувати та швидко знайти необхідну інформацію.
- *Імпорт та експорт даних*. Можливість імпортувати дані автентифікації у Excel-таблицю. Це спростить процес перенесення даних між різними пристроями або резервного копіювання.
- *Захист даних*. Забезпечення високого рівня безпеки та шифрування для збережених даних автентифікації. Помічник використовує сучасні алгоритми шифрування для забезпечення конфіденційності та запобігання несанкціонованому доступу до даних.

Архітектура розробки автономного помічника для збереження даних автентифікації (рис.1) починається зі взаємодії користувача. Користувач взаємодіє з інтерфейсом Telegram-клієнту, яким може бути мобільний додаток, веб-сайт або десктопний додаток. Клієнти надсилають запити до Telegram-сервера за допомогою HTTP-протоколу. Telegram-сервер обробляє запити та надсилає відповіді назад до клієнта. Відповіді можуть містити повідомлення від чат-бота, сповіщення або інші додаткові дані, необхідні для відображення користувачу. Коли нове повідомлення надходить до чат-бота, Telegram-сервер відправляє спеціальний запит - вебхук (webhook), до системи, на якій розташований автономний помічник для збереження даних автентифікації. При встановленні вебхука, система надає Telegram-серверу URL-адресу, на яку треба відправляти вебхук-запити. Коли нове повідомлення надходить до чат-бота, Telegram-сервер автоматично відправляє POST-запит на вказану URL-адресу системи з вмістом повідомлення.

Веб-сервер Nginx виступає в ролі посередника між Telegram-сервером і серверним додатком. При отриманні POST-запиту від Telegram-сервера, веб-сервер Nginx перевіряє правильність URL-адреси та налаштування, що були встановлені під час налаштування вебхука. Після цього Nginx передає отриманий запит до серверного додатка, розробленого на основі фреймворку Lumen, що обробляє запити. Використовуючи можливості для маршрутизації та обробки HTTP-запитів, серверний додаток розпізнає тип отриманого запиту і виконує необхідну логіку роботи: може звертатись до бази даних для збереження та отримання інформації, необхідної для обробки повідомлення. Потім, використовуючи Telegram Bot API, серверний додаток відправляє відповідь до Telegram-сервера, після відправлення відповіді надсилає її назад до клієнтів, яким може бути мобільний додаток, веб-сайт або десктопний додаток. Клієнти отримують відповідь у зручному форматі.

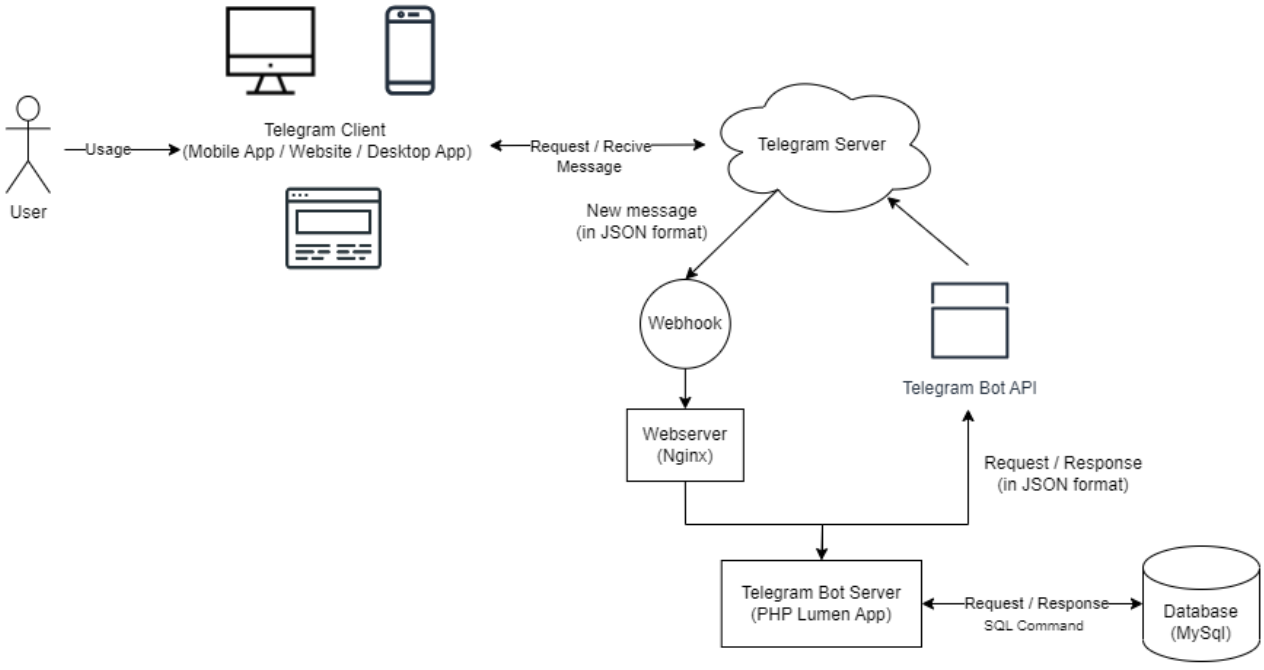

Рисунок 1 - Архітектура автономного помічника для збереження даних автентифікації

Автономний помічник для збереження даних автентифікації передбачає забезпечення високого рівня безпеки. Для досягнення цієї мети важливо враховано безпеку месенджера Telegram та використано метод

безпечного збереження даних:

1. Telegram використовує протокол шифрування MTProto для забезпечення безпеки та конфіденційності комунікації між користувачами та серверами [5]. Цей протокол використовує потокове шифрування, що забезпечує захист від перехоплення повідомлень. Протокол шифрування MTProto розроблений для забезпечення доступу до API сервера з мобільних додатків, виключаючи веб-браузери. Він складається з трьох основних компонентів, що функціонують незалежно один від одного:

— Високорівневий - визначає мову запитів до API, тобто спосіб, яким клієнтські запити та відповіді перетворюються на двійкові повідомлення. Цей компонент відповідає за взаємодію між клієнтом та сервером на рівні повідомлень.

— Криптографічний - відповідає за захист повідомлень шляхом їх шифрування перед передачею через транспортний протокол. Це забезпечує конфіденційність та безпеку даних, що передаються між клієнтом і сервером. Криптографічний рівень забезпечує автентифікацію та авторизацію користувачів, що дозволяє забезпечити безпеку доступу до API.

— Транспортний - визначає протокол передачі повідомлень між клієнтом і сервером. Він використовує існуючий мережевий протокол, такий як HTTP, HTTPS, WS (Simple WebSockets), WSS (WebSockets over HTTPS), TCP або UDP, для забезпечення надійної передачі повідомлень через мережу.

Ці три компоненти працюють разом для забезпечення безпеки та ефективної комунікації між мобільними додатками та API сервером. Протокол MTProto дозволяє забезпечити захист конфіденційної інформації та контролювати доступ до API для забезпечення безпеки та конфіденційності користувачів.

2. Одним з методів безпечного збереження даних автентифікації є використання майстер-ключа. Майстер-ключ є сильним і унікальним ключем, який використовується для захисту важливих даних, таких як паролі до облікових записів. У фреймворку Lumen передбачені методи шифрування та дешифрування даних на основі майстер-ключа. Всі зашифровані значення шифруються за допомогою OpenSSL і шифру AES-256-CBC. Крім того, всі зашифровані значення підписуються кодом автентифікації повідомлення (MAC), щоб виявити будь-які зміни в зашифрованому рядку.

AES-256-CBC (Advanced Encryption Standard 256-bit Cipher Block Chaining) є одним з алгоритмів шифрування, що використовуються для захисту конфіденційності даних [6]. Цей алгоритм використовує ключ довжиною 256 біт та режим шифрування CBC. У режимі CBC, повідомлення розбиваються на блоки фіксованого розміру, наприклад, 128 біт. Кожен блок повідомлення піддається операції XOR з попереднім зашифрованим блоком, а потім шифрується за допомогою ключа AES-256. Цей процес повторюється для кожного блоку повідомлення, що дозволяє досягти ефекту "ланцюжка" (Chaining), де кожен блок залежить від попереднього. AES-256 вказує на використання ключа довжиною 256 біт для шифрування та дешифрування даних. Цей ключ використовується для виконання послідовності раундів, які включають підстановку, перестановку та додавання раундового ключа для кожного блоку.

CBC-режим шифрування додає додатковий рівень безпеки, оскільки він залежить від попереднього зашифрованого блоку. Це унеможливлює просте внесення змін в один блок, оскільки це призведе до неправильного дешифрування наступних блоків.

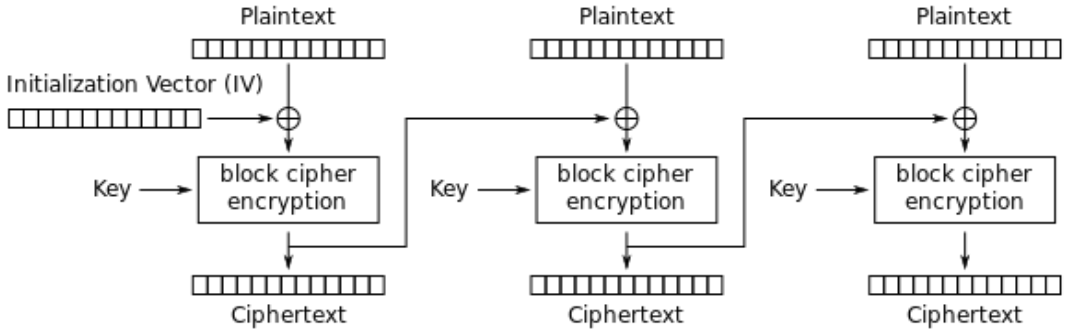

Рисунок 2 - Схема CBC-режим шифрування

AES-256-CBC є складним алгоритмом шифрування, який широко використовується для захисту конфіденційності даних у багатьох сферах, включаючи інформаційну безпеку, комунікації та зберігання даних.

Інформаційна модель автономного помічника, що зберігає дані автентифікації на основі телеграм-боту, передбачає високий рівень безпеки. Це досягається завдяки використанню безпечного шифрування та протоколу MTProto, що забезпечує конфіденційність та безпеку комунікації. Використання майстер-ключа та шифрування AES-256-CBC гарантує захист важливих даних та забезпечує безпеку доступу до системи.

Під час розробки автономного помічника для збереження даних автентифікації проводилось ручне тестування Telegram чат-боту для перевірки його працездатності. Тестування системи здійснювалося шляхом виконання заздалегідь визначених тест-кейсів з урахуванням поставлених вимог до розробки.

Тестування відбувалось за допомогою ноутбуку Lenovo IdeaPad 3 14itl6 12/256 GB.

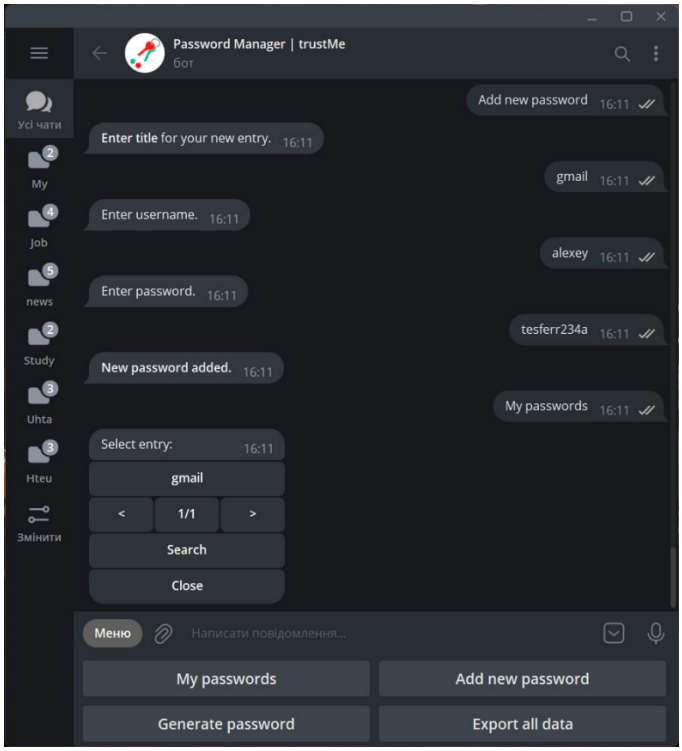

Рисунок 3 - Демонстрація створення даних автентифікації в системі чат-боту

Висновок. Автоматизація в сучасному світі стає все більш важливою для ефективної роботи компаній у різних галузях. Одним з ключових аспектів автоматизації є збереження даних автентифікації, які використовуються для ідентифікації та автентифікації користувачів у системах та програмах. Збереження даних автентифікації відіграє важливу роль у забезпеченні безпеки та захищеності інформації. У результаті була створена та протестована ефективна система автономного помічника, що забезпечує користувачам безпеку та надійність зберігання даних автентифікації та керування своїми обліковими записами та паролями без необхідності запам'ятовувати їх або використовувати однакові паролі для різних сервісів, а також використання чат-бота дає можливість зручного та швидкого доступу до даних з будь-якого пристрою, де є доступ до Інтернет-мережі.

Література

1. Wu TY. Rotating behind security: an enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture / TY. Wu, F. Kong, Q. Meng, S. Kumari, Ch.-M. Chen // EURASIP Journal on Wireless Communications and Networking. - 2023. – 36 (2023).
2. Panchal G. Designing Secure and Efficient Biometric-Based Secure Access Mechanism for Cloud Services / G. Panchal, D. Samanta, A. K. Das, N. Kumar, K. K. R. Choo // IEEE Transactions on Cloud Computing. - 2020.
3. Pearman S. Why people (don't) use password managers effectively / S. Pearman, S. A. Zhang, L. Bauer, N. Christin, L. F. Cranor // In Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security (SOUPS'19). - USENIX Association, USA. - 2019. – p. 319–338.
4. Li Zh. The emperor's new password manager: security analysis of web-based password managers / Zh. Li, W. He, D. Akhawe, D. Song // In Proceedings of the 23rd USENIX conference on Security Symposium (SEC'14). - USENIX Association, USA. - 2014. – p. 465–479.
5. Sušánka T. Security Analysis of the Telegram IM / T. Sušánka, J. Kokeš // In Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium (ROOTS). - Association for Computing Machinery, New York, NY, USA. - 2017. – p. 1–8.
6. Sinurat S. Text EncodingUsing Cipher Block Chaining Algorithm / S. Sinurat, M. Pasaribu // Jurnal Info Sains: Informatikan dan Sains. – 2021. – Vol. 11.

References

1. Wu TY. Rotating behind security: an enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture / TY. Wu, F. Kong, Q. Meng, S. Kumari, Ch.-M. Chen // EURASIP Journal on Wireless Communications and Networking. - 2023. – 36 (2023).
2. Panchal G. Designing Secure and Efficient Biometric-Based Secure Access Mechanism for Cloud Services / G. Panchal, D. Samanta, A. K. Das, N. Kumar, K. K. R. Choo // IEEE Transactions on Cloud Computing. - 2020.

3. Pearman S. Why people (don't) use password managers effectively / S. Pearman, S. A. Zhang, L. Bauer, N. Christin, L. F. Cranor // In Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security (SOUPS'19). - USENIX Association, USA. - 2019. – p. 319–338.
4. Li Zh. The emperor's new password manager: security analysis of web-based password managers / Zh. Li, W. He, D. Akhawe, D. Song // In Proceedings of the 23rd USENIX conference on Security Symposium (SEC'14). - USENIX Association, USA. - 2014. – p. 465–479.
5. Sušánka T. Security Analysis of the Telegram IM / T. Sušánka, J. Kokeš // In Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium (ROOTS). - Association for Computing Machinery, New York, NY, USA. - 2017. – p. 1–8.
6. Sinurat S. Text EncodingUsing Cipher Block Chaining Algorithm / S. Sinurat, M. Pasaribu // Jurnal Info Sains: Informatikan dan Sains. – 2021. – Vol. 11.

The article deals with the topical issue of developing an autonomous assistant for storing authentication data, which will help ensure the security and reliability of access to confidential information and prevent possible data theft. Since, due to the increase in the use of online services and e-commerce, the risk of unauthorized access to users' personal data increases significantly. To implement the development, the PHP programming language, the Lumen framework, the Telegram Bot API, Linux, Git / GitHub were used, MySQL was used as a DBMS. The MTProto protocol used by the Telegram messenger, as well as the encryption methods used in the Lumen framework, namely the methods of data encryption and decryption based on the master key, are considered, which made it possible to ensure the security of transmission and storage of authentication data. All encrypted values are encrypted using OpenSSL and an AES-256-CBC cipher, additionally all encrypted values are signed with a Message Authentication Code (MAC) to detect any changes to the encrypted string. The developed standalone authentication data storage assistant has the following functionality: user authentication, password management, quick search, data import and export, data protection. It also has an important advantage over existing means of data storage, such as storage on a local device, the use of cloud services, the use of password managers, since it provides a high level of protection of user confidential information using modern encryption technologies and reduces the risk of personal data leakage when using online services. After the implementation of the system, testing was carried out and the interaction of the autonomous assistant with the user was demonstrated.
Keywords: *autonomous assistant, encryption algorithm, security, storage of authentication data, chatbot.*

Деркач М.В. – к.т.н., доц, доцент кафедри комп’ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, доцент кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя, e-mail: gln459@gmail.com

Мишко О.Є. – здобувач вищої освіти Східноукраїнського національного університету імені Володимира Даля, e-mail: alex.mishko12@gmail.com