гібридну ML+OPA архітектуру, в якій ML надає ймовірнісні ознаки, а OPA залишається єдиним авторитетним джерелом рішень.[5]

1. OWASP Foundation. OWASP Top 10 for Large Language Model Applications, v2.0. 2025. URL: https://owasp.org/www-project-top-10-for-large-language-model-applications/
2. Regulation (EU) 2024/1689 of the European Parliament and of the Council (EU AI Act). Official Journal of the European Union, 2024.
3. Vakhula O., Opirskyy I. AI Development Security as Code (AISaC): A Policy-Based Approach for Securing AI Engineering Pipelines. CEUR Workshop Proceedings, Vol. 4024, 2025, pp. 170–185.
4. Open Policy Agent. Policy Language (Rego). URL: https://openpolicyagent.org/docs/policy-language (дата звернення: 25.04.2026).
5. Fernández Saura P. et al. On Automating Security Policies with Contemporary LLMs. arXiv:2506.04838, 2025.

Розробка архітектури захищеного менеджера облікових даних з підвищеною стійкістю до GPU-атак

УДК 004.056

Михайло Вдовін[1], Олена Головачова[2]

Національний університет «Одеська політехніка», 9650041@stud.op.edu.ua[1], holovachova@op.edu.ua[2]

Актуальність теми роботи. Зростання кількості вебсервісів, онлайн-платформ та інших цифрових ресурсів призводить до необхідності створення та використання великої кількості облікових даних. У результаті користувачі часто застосовують слабкі або однакові паролі для різних сервісів, що значно підвищує ризик несанкціонованого доступу до їх даних. Менеджери облікових даних є надійним рішенням для централізованого, безпечного зберігання великої кількості облікових даних користувача.

Метою роботи є розробка архітектури захищеного локального менеджера облікових даних, який забезпечує надійне зберігання шляхом використання криптографічних алгоритмів AES-GCM та Argon2.

Однією з найнебезпечніших загроз є офлайн-перебір пароля після отримання файлу з даними або резервної копії. У такому сценарії зловмисник може використовувати потужні графічні процесори (GPU) або спеціалізованих схем (ASIC) для перебору великої кількості варіантів.

Для протидії цьому типу атак в архітектурі програми доцільно застосовувати функції виведення ключів, стійкі до апаратного прискорення. Найбільш розповсюдженим рішенням є Argon2 [1], який спеціально розроблено з урахуванням загроз, пов'язаних з GPU та ASIC.

Менеджер облікових даних будується за такою логікою: 1) Користувач створює головний пароль; 2) Пароль не зберігається напряму, а обробляється алгоритмом Argon2 (створюється хеш для перевірки введеного пароля при вході

та створюється ключ для шифрування даних); 3) Усі дані користувача шифруються та зберігаються в зашифрованому вигляді. Майстер-пароль користувача проходить через Argon2 → отримується ключ шифрування → цим ключем через AES-GCM шифрується база паролів.

На відміну від класичних швидких хешів, Argon2 навмисно сповільнює обробку кожного пароля. Для користувача це означає незначне збільшення часу входу, а для зловмисника – різке зростання очікування при кожній спробі підбору. Саме така асиметрія і є основою криптографічного захисту.

GPU-атаки базуються на здатності графічних процесорів виконувати велику кількість однотипних операцій паралельно. Саме тому зловмисник може перевіряти мільйони варіантів за короткий час, якщо алгоритм хешування є недостатньо складним.

Архітектура менеджера облікових даних розділена на чотири окремі модулі: 1) модуль автентифікації – при першому вході відповідає за створення майстер-пароля, перевірку його надійності та створює файл автентифікації, в якому зберігаються параметри KDF алгоритму і хеш пароля. В подальшому виконує функції перевірки введеного пароля та його перетворення на криптографічний ключ; 2) модуль збереження даних – забезпечує зчитування введених користувачем даних, їх шифрування та збереження у файл; 3) модуль завантаження даних – відповідає за зчитування та розшифрування даних зі сховища; 4) модуль генерації паролів – створює паролі, відповідно до заданих користувачем параметрів. Архітектура наведена на рис.1.

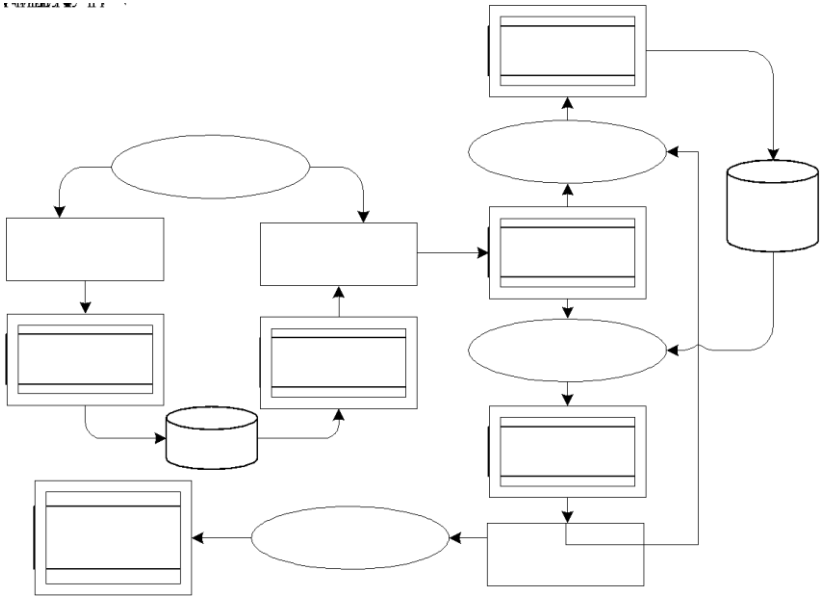

Рис. 1. Архітектура менеджера облікових даних

Argon2 знижує ефективність таких атак завдяки трьом властивостям:1) Argon2 є алгоритмом [2], який вимагає значного обсягу оперативної пам'яті для обчислення. GPU мають високу обчислювальну потужність, але пам'ять для кожного потоку обмежена. Якщо алгоритм потребує багато пам'яті на одну спробу, то кількість одночасних перевірок різко зменшується; 2) Хоча GPU добре підходять для задач із незалежними обчисленнями, Argon2 формує залежності між блоками пам'яті так, що атака стає менш ефективною; 3) Argon2 дозволяє регулювати обсяг пам'яті кількість проходів, ступінь паралелізму. Це дозволяє підібрати параметри, при яких вхід для користувача залишиться зручним, але GPU-атака стане не вигідною.

Запропонована архітектура поєднує зручність для користувача та високий рівень криптографічного захисту.

1. Argon2: вебсайт – URL: https://www.argon2.com (дата звернення: 11.04.2026).
2. Biryukov A., Dinu D., Khovratovich D., Josefsson S. RFC 9106 – Argon2 memory-hard function for password hashing and proof-of-work applications. – 2021.

Machine learning methods for automated assessment in distance learning

UDK 004.89:37.018.43

Kostiantyn Radchenko[1], Wei Shenlai[2]

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", [1]radchenko.kostiantyn@lll.kpi.ua, [2]radche001@gmail.com

The rapid transition to distance education has intensified the need for automated assessment technologies capable of supporting instructors in evaluating large volumes of student work [1]. In many learning management systems (LMS), assessment remains predominantly manual, requiring significant time and leading to potential subjectivity in scoring. Machine learning (ML) methods provide new opportunities for automating evaluation processes and generating individualized feedback for learners [2]. Unlike rule-based approaches, ML models can recognize linguistic patterns, semantic relationships, and contextual nuances in student responses. This study examines the application of machine learning methods for automated scoring and real-time feedback generation in LMS environments, with particular attention to adaptability, interpretability, and efficiency.

One of the most challenging aspects of distance learning is the evaluation of open-ended student responses [3]. Manual assessment demands substantial instructional resources and may result in inconsistent scoring across different evaluators. Existing automated systems typically focus on multiple-choice or short-answer items that can be assessed using pattern-matching techniques. However, such approaches cannot adequately capture semantic depth, logical structure, or argumentative quality in extended written responses. The central objective of this work is the development of