УДК: 004.056.5:004.421

БАГАТОРІВНЕВИЙ ЗАХИСТ КРИПТОГРАФІЧНИХ КЛЮЧІВ НА ОСНОВІ PBKDF2 ТА AES-256 CBC

Пилипенко В.І. старший викладач
Київський національний університет технологій та дизайну
Горбенко М.О., студент
Київський національний університет технологій та дизайну
Лисенко А.С., студентка
Київський національний університет технологій та дизайну

Ключові слова: криптографічні ключі, PBKDF2, AES-256, CBC, інформаційна безпека, key derivation, Python, C#.

Захист криптографічних ключів є фундаментальним завданням сучасних інформаційних систем, оскільки компрометація ключового матеріалу призводить до втрати конфіденційності та цілісності захищених даних. На відміну від звичайної інформації, криптографічні ключі мають критичну цінність, адже забезпечують функціонування механізмів шифрування, автентифікації та контролю доступу[1-3]. Загрози безпеці виникають як ззовні, так і всередині системи, включаючи вразливості програмного забезпечення, помилки конфігурації, наслідки дії шкідливого програмного забезпечення та людський фактор. У зв'язку з цим використання примітивних підходів до зберігання ключів у відкритому вигляді є неприйнятним, а застосування стандартизованих криптографічних механізмів є необхідністю. Зокрема, широко використовуються PBKDF2 та AES-256, які забезпечують багаторівневий захист ключового матеріалу та даних.Алгоритм PBKDF2 (Password-Based Key Derivation Function 2) використовується для виведення криптографічного ключа з пароля шляхом багаторазового застосування криптографічної хеш-функції з використанням випадкової солі та великої кількості ітерацій [4]. Це суттєво ускладнює атаки типу brute force та словникові атаки, оскільки кожна спроба підбору пароля потребує значних обчислювальних ресурсів. Симетричний алгоритм AES-256 у режимі CBC (Cipher Block Chaining) забезпечує конфіденційність даних шляхом послідовного шифрування блоків, де кожен наступний блок залежить від попереднього шифротексту, що дозволяє приховувати структурні закономірності відкритого тексту та підвищує стійкість до криптоаналізу [5]. Поєднання PBKDF2 та AES-256 CBC формує багаторівневу модель захисту, у якій пароль користувача не використовується безпосередньо як ключ, а проходить етап криптографічної деривації, що суттєво знижує ризики компрометації навіть у разі часткового витоку даних [6].

Для реалізації запропонованого підходу використано мову програмування Python. В програмному лістингу 1 наведено фрагмент реалізації генерації ключа та вимірювання часу виконання:

*Програмний лістинг 1.*Фрагмент реалізації генерації ключа та вимірювання часу виконання:

```
import time
import hashlib
import os
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad

# функція генерації ключа через PBKDF2
def derive_key(password, salt, iterations):
    return hashlib.pbkdf2_hmac(
        'sha256',
        password.encode(),
        salt,
        iterations,
        dklen=32  # 256 біт
    )

# тестування швидкодії
def benchmark():
    password = "secure_password"
    salt = os.urandom(16)

    iterations_list = [1, 1000, 10000, 100000]
    results = []

    for iters in iterations_list:
        start = time.time()
        key = derive_key(password, salt, iters)
        end = time.time()

        results.append((iters, end - start))

    return results

# запуск експерименту
data = benchmark()

for iters, t in data:
    print(f"Iterations: {iters}, Time: {t:.6f} sec")
```

Також реалізовано шифрування даних за допомогою AES-256 у режимі CBC, яке наведено в програмному лістингу 2:

*Програмний лістинг 2.*Шифрування даних за допомогою AES-256 у режимі CBC

```
defencrypt_data(key, plaintext):
iv = os.urandom(16)
cipher = AES.new(key, AES.MODE_CBC, iv)

ciphertext = cipher.encrypt(pad(plaintext.encode(),
AES.block_size))
returniv + ciphertext

# прикладвикористання
key = derive_key("secure_password", os.urandom(16), 100000)
encrypted = encrypt_data(key, "Sensitivedata")
```

Експериментальне дослідження було спрямоване на оцінку впливу кількості ітерацій PBKDF2 на продуктивність системи. Результативимірюванняпродуктивностіпоказали:

- SHA-256 (1 ітерація) – 0.00049 с, рівень захисту низький
- PBKDF2 (1 000 ітерацій) – 0.00025 с, рівень захисту середній
- PBKDF2 (10 000 ітерацій) – 0.0023 с, рівень захисту високий
- PBKDF2 (100 000 ітерацій) – 0.0205 с, рівень захисту дуже високий

Для візуалізації результатів побудовано графік залежності часу виконання від кількості ітерацій PBKDF2, який демонструє майже логарифмічне зростання обчислювальної складності зі збільшенням параметра ітерацій. Це підтверджує властивість key-stretching, згідно з якою підвищення криптографічної стійкості досягається за рахунок збільшення обчислювальних витрат. Дані було зібрано за рахунок програмного сервісу написаного на мові програмування C#, а графік побудовано з використанням бібліотеки matplotlib та представлено на рис. 1.

Отримані результати показують, що збільшення кількості ітераційPBKDF2 суттєвопідвищуєстійкістьдоатак, однаксупроводжуєтьсязростаннямчасуобчислень. Таким чином, у практичних системах необхідно забезпечувати баланс між рівнем безпеки та продуктивністю. Поєднання PBKDF2 та AES-256 CBC дозволяє реалізувати ефективну багаторівневу систему захисту криптографічних ключів, яка відповідає сучасним вимогам інформаційної безпеки та може бути застосована у серверних і прикладних системах, де критично важливим є захист секретних даних.

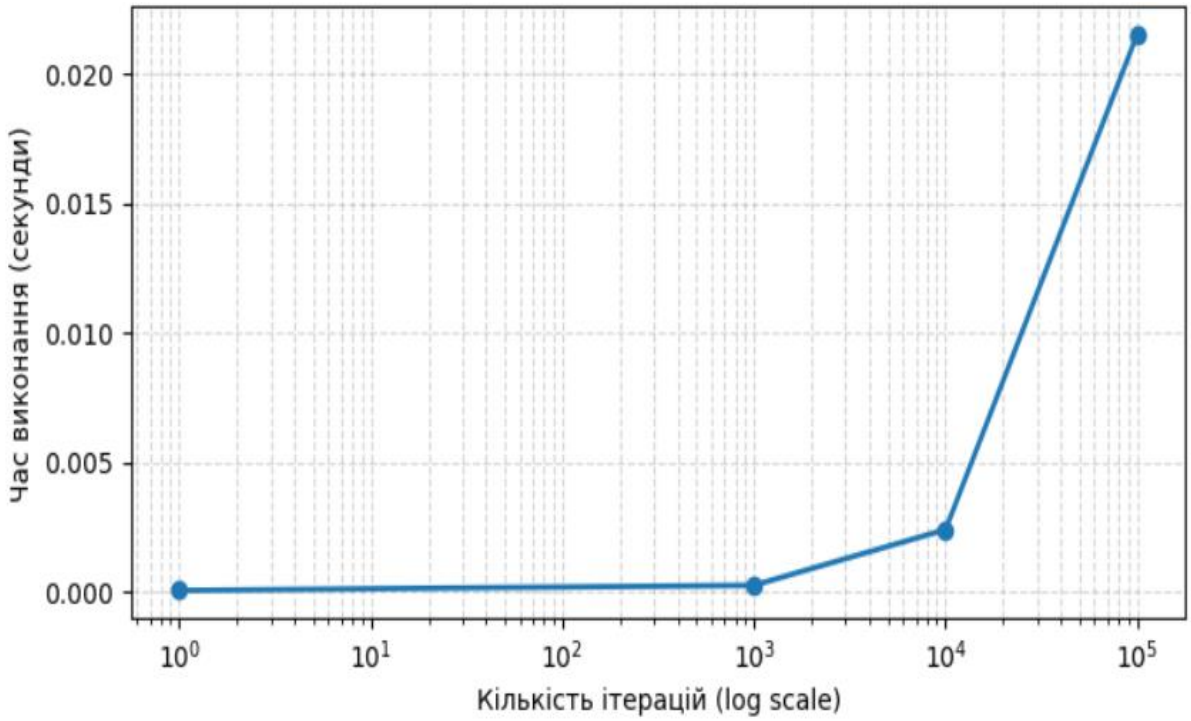

Рис. 1 Залежність часу виконання PBKDF2

Список використаних джерел

1. Turan, M. S., Barker, E., Burr, W., & Chen, L. (2010). Recommendation for password-based key derivation. NIST special publication, 800(132), 42.

2. Aumasson, J. P. (2024). Serious cryptography: a practical introduction to modern encryption. No Starch Press, Inc.

3. Huang, D., Chowdhary, A., & Pisharody, S. (2018). Software-Defined networking and security: from theory to practice. CRC press.

4. Iuorio, A. F., & Visconti, A. (2018, August). Understanding optimizations and measuring performances of PBKDF2. In International Conference on Wireless Intelligent and Distributed Environment for Communication (pp. 101-114). Cham: Springer International Publishing.

5. Rao, S., Mahto, D., Yadav, D. K., & Khan, D. (2017). The AES-256 cryptosystem resists quantum attacks. Int. J. Adv. Res. Comput. Sci, 8(3), 404-408.

6. Santos, P., Cunha, L., Soares, A., Váz, P., Silva, J., Martins, P., & Abbasi, M. (2025, July). Privacy-Preserving Healthcare Analytics: A Hybrid Approach Using AES-256-CBC Encryption and Differential Privacy in Mobile Applications. In International Conference on Disruptive Technologies, Tech Ethics and Artificial Intelligence (pp. 355-367). Cham: SpringerNatureSwitzerland.