DOI 10.15276/imms.v12.no4.306
УДК 681.514:621.029

Informatics and Mathematical Methods in Simulation
Vol. 12 (2022), No. 4, pp. 306-314

# РОЗРОБКА ДОДАТКУ ДЛЯ ШИФРУВАННЯ ДАНИХ ЗІ ЗБЕРЕЖЕННЯМ ФОРМАТУ

Т.І. Горбатюк, О.Ю. Лебедєва

Національний університет «Одеська Політехніка», просп. Шевченка, 1, Одеса, 65044, Україна; e-mail: infsec2011@gmail.com

Розглядається розробка додатку для шифрування даних зі збереженням формату. Наводяться основні визначення, такі як особиста інформація, конфіденційна ідентифікаційна інформація. За останні кілька років зростання кількості витоків персональних даних призвело до втрати цілісності даних для сотень мільйонів записів користувачів. Такі атаки спрямовані як проти великих компаній, так і проти малих підприємств. Частка кібератак з метою крадіжки персональних даних у підприємств забезпечена через халатність компаній щодо коректного шифрування та маскування персональних даних працівників та клієнтів. У роботі розкривається поняття шифрування даних із збереженням формату, як процес шифрування даних таким чином, щоб вихідні дані залишалися в тому самому форматі, що й вхідні дані. У шифруванні із збереженням формату розглядається два режиму роботи – FF1 та FF3. При аналізі існуючих режимів шифрування зі збереженням формату, був виділений режим FF3-1, який був спеціально створений для вирішення проблеми шифрування на невеликих доменах тобто областях даних. Режим FF3-1 був реалізований у програмному додатку. У роботі також розкривається поняття псевдонімізації та методи досягнення її. Псевдонімізація може бути досягнута за допомогою різних методів, таких як маскування даних, шифрування або токенізація. Процеси маскування даних змінюють значення даних залишаючи їх вхідний формат. Мета полягає в тому, щоб створити версію, яка не піддається розшифровці або зворотній інженерії. В роботі було обрано метод маскування даних. У зв'язку з цим основним завданням є втілити методи які б допомагали компаніям шифрувати та маскувати дані з мінімальними вкладами в реструктуризацію та збереження функціональності на льоту. Для того, щоб досягти цілі чіткої псевдонімізації, щоб факт підміни реальних даних був непомітним, вирішено модернізувати підхід до маскування та шифрування для полів типу ім’я, прізвище та email адреса. В роботі удосконалено методи псевдонімізації та маскування полів даних: типу ім’я, прізвище та email адреса та застосування методу FPE в режимі FF3-1 для полів типу номер телефону та кредитної карти. Удосконалений метод псевдонімізації та маскування полів даних реалізовано у програмному додатку.

**Ключові слова**: персональна інформація, шифрування зі збереженням формату, псевдонімізація, маскування даних.

## Вступ

Особиста інформація (Personal Identifiable Information PII) – це будь-які дані, які можуть бути використані для ідентифікації конкретної особи. Приклади включають повне ім’я особи, телефонні та номери соціального страхування, фізичні або імейл адреси, тощо.

Особиста інформація містить не лише очевидні посилання на особистість людини, а і інформаційні фрагменти, які в поєднанні з іншими наборами даних розкривають особу, тобто класифікується як ідентифікаційна інформація.

Конфіденційна ідентифікаційна інформація включає будь-який набір даних, який містить повне ім’я, адресу або фінансову інформацію. Неконфіденційна ідентифікаційна інформація – це будь-які загальні дані, доступні з загальнодоступних ресурсів (таких як профілі в соціальних мережах), наприклад поштовий індекс або дата народження.

Доступ до ідентифікаційної інформації без авторизації становить значний ризик як для окремих осіб, так і для компанії. Індивідуальна шкода може включати крадіжку особистих даних, шахрайство або шантаж. Водночас компанія може зазнати шкоди через порушення цілісності даних, наприклад втрати довіри громадськості, юридичної відповідальності або великих штрафів.

Ідентифікаційну інформацію можна оцінити шляхом визначення рівня її впливу на конфіденційність ідентифікаційної інформації. Рівні впливу на конфіденційність ідентифікаційної інформації варіюються від низького, помірного або високого, щоб вказати на потенційну шкоду, яка може бути завдана особі чи організації, якщо дані будуть скомпрометовані.

Під час внутрішніх тестів на проникнення в мережу консультанти з безпеки часто знаходять конфіденційну інформацію, яка незахищено зберігається на файлових серверах. Це включає чутливі дані про особу (повне ім’я, телефони, адреси, як фізичні так і електронні), номери кредитних карток, паспортів, сторонніх додатків для входу в платіжні системи, аналітику та інші веб-маркетингові та бізнес-портали. Це один із головних ризиків, виявлених під час незахищених практик зберігання інформації.

Оскільки кожне підприємство, незалежно від його розміру, підлягає під закони про ідентифікаційну інформацію, нормативні акти та інші повноваження, пов’язаних із захистом ідентифікаційної інформації, воно повинне дотримуватися чіткого підходу щодо безпеки та конфіденційності даних, щоб захистити ідентифікаційну інформацію, яку зберігає і використовує у своєму середовищі [1]. Часто перепоною для компаній, що оперують персональними даними людей є комплексність в реалізації шифрування даних, які вони зберігають. Це пов’язане з тим, що часто вони використовують застарілі системи, чиї кодові бази та моделі даних не можна змінити або надто обтяжливі та ризиковані для оновлення. Наприклад, база даних, яка зберігає конфіденційні дані про охорону здоров’я, яка була введена в дію майже 20 років тому і продовжує працювати. Більшість організацій схильні до ризиків, пов’язаних з реструктуризацією такої ключової виробничої системи [2].

Постійно зростаюча кількість випадків витоку даних, пов’язаних із особистою (ідентифікаційною) інформацією, призвела до втрат акціонерів на мільярди доларів, штрафів на мільйони доларів і збільшення ризику крадіжки особистих даних осіб, чиї конфіденційні дані були розкриті. Порушення безпеки даних є небезпечним як для окремих осіб так і організацій. У зв'язку з цим актуальним завданням є втілити методи які б допомагали таким компаніям шифрувати та маскувати дані з мінімальними вкладами в реструктуризацію та збереження функціональності на льоту.

**Мета статті та постановка задач**

Метою є розробка додатку для шифрування персональних даних зі збереженням формату шляхом удосконалення методу псевдонімізації та маскування даних.

Для досягнення поставленої мети необхідно розв’язати наступні задачі:

1. Проаналізувати принципи роботи та визначити переваги та недоліки сучасних методів шифрування зі збереженням формату та методи псевдонімізації даних.

2. Удосконалити метод псевдонімізації та маскування даних.

3. Реалізувати додаток для шифрування даних зі збереженням формату на основі розробленого покращеного методу псевдонімізації та маскування даних.

Об’єктом дослідження є удосконалення методу псевдонімізації та маскування даних.

Предметом дослідження є методи шифрування зі збереженням формату та

псевдонімізації даних.

Отримані результати можуть бути використані для підвищення стійкості до кібератак підприємств та покращення захисту персональних даних, якими вони користуються і зберігають.

**Основна частина**

Шифрування із збереженням формату (FPE) – це процес шифрування даних таким чином, щоб вихідні дані (зашифрований текст) залишалися в тому самому форматі, що й вхідні дані (відкритий текст). Значення «формату» різне. Зазвичай використовуються лише кінцеві набори символів; цифровий, буквений або буквено-цифровий» [3].

FPE дає змогу розгортати широко поширене шифрування без критичних наслідків для застарілих систем. Додаткам, яким не потрібен доступ до конфіденційних даних, надаються шифртексти FPE. Якщо програмі потрібен доступ до даних, у робочий процес можна вставити функцію дешифрування, щоб надати доступ до відкритого тексту.

У шифруванні із збереженням формату розглядається три режими роботи – FF1, FF2 та FF3, які в цілому називаються FFX. (NIST «Спеціальна публікація (SP) 800-38G, Рекомендації щодо режимів роботи блокового шифру: методи шифрування зі збереженням формату»).

FFX використовує кілька раундів функції Feistel над відкритим текстом разом із ключем для створення зашифрованого тексту. Функція Feistel розділяє відкритий текст на дві частини, переставляє текст, щоб змінити його вигляд, а потім змінює ліву половину тексту на праву і навпаки. Метод FF1 використовує 10 раундів функції Фейстеля, а FF3 використовує 8 раундів, FF2 так і не отримав схвалення NIST.

FF3-1 – це схема шифрування, що настроюється, створена для вирішення проблеми шифрування на невеликих доменах. Було введено налаштування (tweaks) для синтетичного збільшення доменного простору. З введенням налаштувань зловмисник тепер повинен мати правильне шифрування відкритого тексту та правильне налаштування, щоб через кодову книгу повернутися від шифрування до відкритого тексту.

При аналізі існуючих режимів шифрування зі збереженням формату, був виділений режим FF3-1 [4], який був реалізований у програмному додатку.

У європейскому загальному регламенту захисту даних (GDPR) псевдонімізація визначається як «обробка персональних даних таким чином, що дані більше не можуть бути пов’язані з певним суб’єктом даних без використання додаткової інформації». Таким чином відбувається обмін особистими даними з неідентифікуючими даними, і для відтворення вихідних даних потрібна додаткова інформація.

Метод псевдонімізації, як визначено в GDPR, – це будь-який метод, який гарантує, що дані не можна використовувати для ідентифікації особистості. Це вимагає видалення прямих ідентифікаторів і, бажано, уникнення кількох ідентифікаторів, які в поєднанні можуть ідентифікувати особу. Крім того, ключі шифрування або інші дані, які можна використовувати для повернення до початкових значень даних, слід зберігати окремо та надійно.

Псевдонімізація може бути досягнута за допомогою різних методів, таких як маскування даних, шифрування або токенізація. Вона зазвичай використовується як техніка для захисту особистих даних у застарілих виробничих системах від несанкціонованого доступу, де інші методи безпеки непридатні.

Маскування даних – це спосіб створити фальшиву, але реалістичну версію організаційних даних. Мета полягає в тому, щоб захистити персональні та конфіденційні дані, одночасно забезпечуючи функціональну альтернативу, коли

реальні дані не потрібні, наприклад, під час навчання користувачів, демонстрацій продажів або тестування програмного забезпечення.

Процеси маскування даних змінюють значення даних залишаючи їх вхідний самий формат. Мета полягає в тому, щоб створити версію, яка не піддається розшифровці або зворотній інженерії. Є кілька способів змінити дані, включаючи перетасування символів, заміну слів або символів і шифрування (рисунок 1).

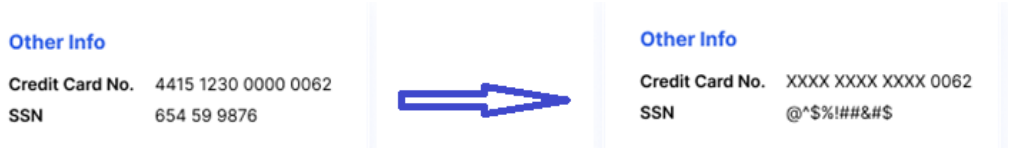


**Рис.1.** Приклад маскування даних

Існує чимало методів маскування даних. Розглянемо такі як перетасування, обнуління, варіація значень та перестановка.

Якщо існує, ідентифікаційний номер, такий як 76498 у робочій базі даних, можна замінити на 84967 у тестовій базі даних. Це метод перетасовування окремих знаків у значенні даних. Він дуже простий у реалізації, але його можна застосувати лише до деяких типів даних і він менш безпечний.

Під час перегляду неавторизованим користувачем дані виглядають відсутніми або «нульовими», це метод обнуління. Але це робить дані менш корисними для розробки та тестування.

Вихідні значення даних замінюються функцією, наприклад різницею між найменшим і найвищим значенням у ряді. Наприклад, якщо клієнт придбав кілька продуктів, ціну покупки можна замінити діапазоном між найвищою та найнижчою сплаченою ціною. Це метод варіації значень і він може надати корисні дані для багатьох цілей, не розкриваючи вихідний набір даних.

В методі перестановки, за винятком того, що значення даних змінюються в межах одного набору даних. Дані переставляються в кожному стовпці за допомогою випадкової послідовності; наприклад, перемикання між реальними іменами клієнтів у кількох записах клієнтів. Вихідний набір виглядає як реальні дані, але він не відображає справжню інформацію для кожної особи чи запису даних.

Був проведений аналіз продуктів з імплементованим методами псевдонімізації [5] та шифруванням зі збереженням формату та виділені їх сильні та слабкі сторони. Після аналізу сильних та слабких сторін продуктів на ринку, було виявлено, що реалізовані продукти пропонують лише типи маскування за допомогою спеціальних знаків (“*”, “#”, або “0”) та шифрування полів типу ім’я, прізвище та email адреса в рамках змішування значень в межах документа, або перемішування букв в межах самих значень, що слабо підходить під загальну концепцію псевдонімізації даних. Тому було вирішено покращити методи псевдонімізації та маскування полів даних: типу ім’я, прізвище та email адреса та розумне застосування методу FPE в режимі FF3-1 для полів типу номер телефону та кредитної карти.

Для того, щоб досягти цілі чіткої псевдонімізації, щоб факт підміни реальних даних був непомітним, вирішено модернізувати підхід до маскування та шифрування для полів типу ім’я, прізвище та email адреса.

Для псевдонімізації поля “ім’я” та “прізвище” потрібно замінювати реальні значення на інші, яке будуть іншими реальними іменами і прізвищами. Щоб виконати поставлену задачу, необхідно створити словники з унікальними людськими (в межах словника) іменами та прізвищами, достатньо великі, щоб алгоритм випадково вибраного числа міг обрати значення за порядковим номером

для подальшої заміни справжнього імені і прізвища користувача. Потім в окремий словник з ключами для імен та прізвищ буде записано реальне ім'я/прізвище користувача (значення) і порядковий номер фейкового імені/прізвища (ключ) для подальшого процесу де-псевдонімізації. Наприклад: якщо за алгоритмом було згенероване число 23, то у словнику з фейковими іменами шукаємо ім'я за порядковим номером 23. Заміняємо реальне ім'я користувача на псевдонім "Jonathan" та зберігаємо реальне ім'я окремо в словник ключів поряд з згенерованим числом 23. Такий же процес для заміни значення прізвища.

Процес де-псевдонімізації буде відбуватись у зворотньому напрямку. Підгружаючи файл із зашифрованими та замаскованими даними, програма розділить файл на колонки і зчитає значення в колонці "ім'я", і знайде значення "Jonathan" у словнику фейкових імен. Відповідно до знайденого імені, за алгоритмом буде взято порядковий номер цього імені у словнику фейкових імен та потім у словнику ключів для імен за ключем порядкового номеру, числа 23, знайдемо реальне ім'я користувача, яке буде значенням у цьому словнику для ключа 23. Такий же процес для де-псевдонімізації поля прізвища. Таким чином, частина де-псевдонімізації полів "ім'я" та "прізвище" буде завершена.

Було також розроблене та імплементоване рішення для захисту словників даних за допомогою AES (режим CBC) з ключем шифрування в 128-біт [6].

Щодо удосконаленого методу псевдонімізації для поля "email" адреси, то процес передбачає відокремлення домену (domain) адреси від частини юзернейму (local-part) користувача, тому що маскувати чи шифрувати домен email адреси частіше за все не є виправданим. Процес псевдонімізації email адреси буде спиратися на замасковані ім'я та прізвище, щоб у вихідному (замаскованому і зашифрованому) датасеті дані виглядали як справжні. Для цього візьмемо першу букву замаскованого імені і ціле замасковане прізвище та поєднаємо їх через крапку. Наприклад, для замаскованого юзера з ім'ям Jonathan та прізвищем Richardson частина юзернейма після алгоритму об'єднання буде виглядати наступним чином: J.Richardson. Наступним кроком буде частина генерації 8 значного випадкового числа (наприклад, 74925207), і приєднання цього номеру до згенерованої частини юзернейму email адреси. Також, це 8 значне згенероване число (74925207) буде служити ідентифікатором, тобто ключем, для зберігання реального юзернейму email адреси як значення у словнику ключів для email адрес для їх подальшої де-псевдонімізації. Тобто, замаскований юзернейм email адреси виглядатиме наступним чином: J.Richardson74925207. Потім до згенерованого юзернейму додається символ @ та домен реальної email адреси і це буде замасковане значення email адреси.

Процес де-псевдонімізації для поля email адреси виглядатиме наступним чином: підгружаючи файл із зашифрованими та замаскованими даними, програма розділить файл на колонки і зчитає значення в колонці "email" адрес, і знайде значення "J.Richardson74925207@gmail.com". Згідно подальших кроків, алгоритм зіскануює значення email адреси і виявить у ній 8-значне число (74925207), потім у словнику ключів для email адрес буде виконано пошук по ключам і для знайденого ключа 74925207 поряд буде значення реального юзернейму email адреси. Наступним кроком стане приєднання (через символ @) незамаскованого домену email адреси до знайденого значення (реального юзернейму) у словнику ключів для email адрес. Таким чином, частина де-псевдонімізації поля "email" адреса буде завершена.

Для повноцінного маскування та шифрування зі збереженням формату для поля "номер телефону" було обрано наступний алгоритм дій. За основу було взято номер телефону українського формату, де можливі два типи збереження з кодом країни (+380) та без нього. Оскільки в табличних даних, де зберігаються номери

телефонів, (в csv файлах) частіше за все, не підтримується форматування для цифрових значень, що починається з 0 або з +, було вирішено не враховувати шифрування таких випадків. А отже, маємо значення довжиною в 12 і 9 символів відповідно.

Для того, щоб забезпечити гарантовано надійне маскування поля номера телефону і 12 символів, було вирішено відділити частину номера телефону, що є кодом країни (2 символа) і також код мобільного оператора (3 символа), тобто, перші 5 символів і не піддавати їх шифруванню, а шифрувати методом зі збереженням формату FPE в режимі FF3-1. Тобто, наприклад, коли програма зчитує мобільний номер з поля у вигляді “380671234567”, то значення “38067” записується як окреме значення і зберігається окремо, а значення “1234567” буде шифруватись і перетворюватись у інше цифрове значення такої ж довжини (7 цифр). Після того до зашифрованого значення приєднуємо значення з кодом країни і кодом оператора і зберігаємо як повноцінне зашифроване і замасковане значення. Під час шифрування методом зі збереженням формату FPE в режимі FF3-1 нам потрібні два значення key та tweak, які будуть збережені у словнику для ключів мобільних телефонів. При розшифруванні береться номер по порядку і проходиться по словнику ключів розшифровуючи значення за значенням, таким чином відбувається процес розшифрування.

Зазвичай, люди не вникають у глибокий аналіз номерів карток. Для них достатньо всього побачити 16 цифр щоб ідентифікувати, що це саме номер картки. Тому було вирішено шифрувати усі 16 цифр картки, щоб уникнути можливих ризиків методом зі збереженням формату FPE в режимі FF3-1. Тобто, наприклад, коли програма зчитує поле “credit card” і дані карток з нього, то шифрування зі збереженням формату буде виконуватись над 16 цифрами, де в результаті шифрування на виході буде інше зашифроване 16-значне значення.

У розробленому програмному додатку пропонується два режиму на вибір для корисувача: зашифрувати, та розшифрувати файли. Вікно при виборі режиму шифрування зображено на рисунку 2, де зліва показано опції розділювача для підвантаженого .csv файлу з реальними даними користувачів, та кнопка підвантаження файлу. Праворуч зображений treeview з підвантаженими колонками з файлу користувача та даними в ньому, а також динамічно згенеровані labels, що відповідають назвам колонок, та опції dropdown lists (рис.3), де наявні назви методів якими користувач може зашифрувати або замаскувати відповідну до label-а колонку.

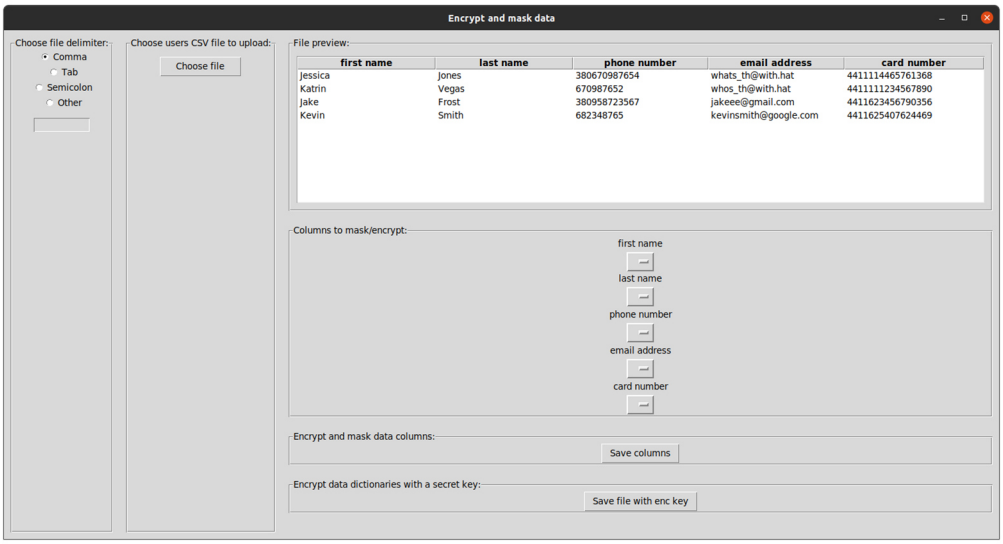


**Рис. 2**. Вікно програми в режимі шифрування

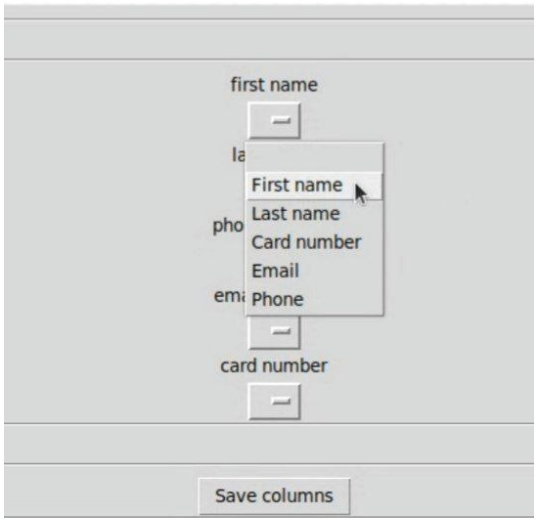


**Рис. 3.** Опції вибору у dropdown list-ах

Потім користувач натискає на кнопку “Save columns” та зберігає encrypted.csv файл разом з файлами словників з ключами відповідно до вибраних опцій в dropdown lists. Наступним кроком користувачу потрібно натиснути кнопку “Save file with enc key”, що зашифрує словники (за допомогою AES (режим CBC) з ключем шифрування в 128-біт) з ключами та збереже один файл ключа під назвою mykey.key для майбутнього розшифрування цих словників.

Для порівняння вхідних і вихідних (замаскованих та зашифрованих даних наведені наступні скріншоти датасетів (рис. 4) та (рис. 5) відповідно.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | first name | last name | phone number | email address | card number |
| 2 | Jessica | Jones | 380670987654 | whats_th@with.hat | 4411114465761368 |
| 3 | Katrin | Vegas | 670987652 | whos_th@with.hat | 4411111234567890 |
| 4 | Jake | Frost | 380958723567 | jakeee@gmail.com | 4411623456790356 |
| 5 | Kevin | Smith | 682348765 | kevinsmith@google.com | 4411625407624469 |
| 6 | | | | | |

**Рис. 4.** Вхідний файл з реальними даними користувачів users_w_comma.csv

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | first name | last name | phone number | email address | card number |
| 2 | Paul | Sanders | 380677766448 | P.Sanders86077122@with.hat | 7698650865550020 |
| 3 | Jeffrey | Conley | 679419326 | J.Conley29851818@with.hat | 6764499424904407 |
| 4 | Sue | Acevedo | 380952469074 | S.Acevedo47794405@gmail.com | 5127538074400576 |
| 5 | Padraig | Kline | 683828986 | P.Kline74270084@google.com | 4794620070515788 |
| 6 | | | | | |

**Рис.5.** Вихідний файл з замаскованими та зашифрованими даними користувачів encrypted.csv

Наступним пунктом є дешифрування та де-псевдонімізація даних. Для цього користувач при запуску програми вибирає опцію розшифрування. У оновленому вікні (рис. 6) він підвантажує попередьо зашифрований файл encrypted.csv та ключ для розшифрування словників, потім натискає на кнопку “Decrypt users data” та після оповіщення про успішне розшифрування натискає на кнопку “Save decrypted file”.

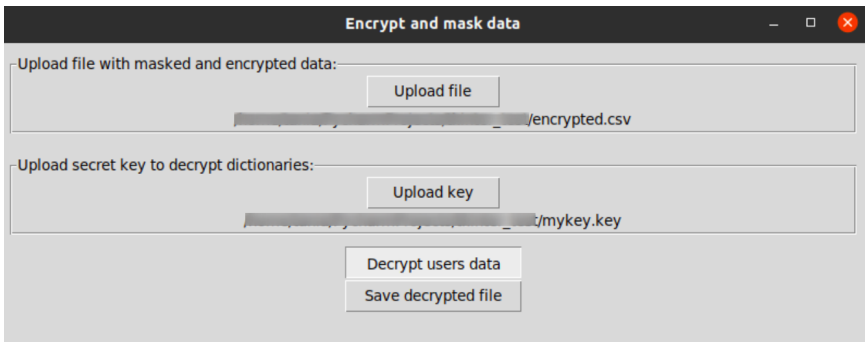


**Рис. 6**. Вікно програми в режимі дешифрування

Після цього файл буде збережено у вибраній директорії і таким чином процес дешифрування та де-псевдонімізації даних завершено.

Метод шифрування зі збереженням формату та псевдонімізації даних у розробленому додатку було реалізовано за допомогою використання наступних бібліотек Python: codecs, string, cryptography.fernet >> Fernet, pandas, csv, random, os, binascii, fff3 >> FF3Cipher, Crypto.Cipher >> AES, numpy, pandas >> read_csv.

**Висновки**

Під час розробки додатку для шифрування персональних даних зі збереженням формату були імплементовані удосконалені методи псевдонімізації та маскування даних. Були розв'язані поставлені на початку задачі та був розроблений додаток для шифрування даних зі збереженням формату та удосконаленими методами псевдонімізації на високорівневій мові програмування Python з використанням допоміжних бібліотек. Розроблене програмне забезпечення задовольняє шифрування персональних даних зі збереженням формату та удосконаленими методами псевдонімізації полів даних: типу ім'я, прізвище та email адреса та імплементоване розумне застосування методу FPE в режимі FF3-1 для полів типу номер телефону та кредитної карти.

Результати даної роботи можуть бути можуть бути використані для підвищення стійкості до кібератак підприємств та покращення захисту персональних даних, якими вони користуються і зберігають.

**Список літератури**

1. PII Protect Cybersecurity. How to Secure your Data URL: https://thecyphere.com/blog/pii-protect/
2. What is format-preserving encryption (FPE) and its benefits? URL: https://www.ubiqsecurity.com/what-is-format-preserving-encryption-fpe-and-its-benefits%EF%BF%BC/
3. Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption URL: https://csrc.nist.gov/publications/detail/sp/800-38g/rev-1/draft
4. Recent Cryptanalysis of FF3 URL: https://csrc.nist.gov/news/2017/recent-cryptanalysis-of-ff3
5. Pseudonymization according to the GDPR [definitions and examples] URL: https://dataprivacymanager.net/pseudonymization-according-to-the-gdpr/
6. Advanced Encryption Standard (AES) URL: https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard

# DEVELOPMENT OF AN APPLICATION FOR DATA ENCRYPTION WITH FORMAT PRESERVATION

T. Horbatiuk, O. Lebedieva

National Odesa Polytechnic University,
1, Shevchenko Ave., Odesa, 65044, Ukraine; e-mail: infsec2011@gmail.com

The paper describes the development of an application for data encryption with format preservation. Basic definitions such as personal information, identifying information are provided. Over the past few years, a growing number of personal data breaches have resulted in the loss of data integrity for hundreds of millions of user records. Such attacks are aimed at both large companies and small businesses. The cyber-attacks aimed at stealing personal data from enterprises and it is ensured due to the negligence of companies regarding the conscious encryption and masking of personal data of employees and customers. The paper reveals the concept of format-preserving data encryption, as the process of encrypting data in such a way that the output data remains in the same format as the input data. Two modes of operation are considered in format-preserving encryption - FF1 and FF3. When analyzing the existing FPE modes with format preservation, the FF3-1 mode was selected due to the ability to solve the problem of encryption on small data domains. FF3-1 mode was implemented in a software application. The work also reveals the concept of pseudonymization and methods of achieving it. Pseudonymization can be achieved using various methods such as data masking, encryption or tokenization. Data masking process change the value of data while leaving its input format. The goal is to create an output that cannot be deciphered or reverse engineered. The method of data masking was implemented. Thus, the main task was to implement methods that would help companies encrypt and mask data with minimal contributions to restructuring and preserving functionality on the fly. In order to achieve the goal of clear pseudonymization, so that the fact of replacing real data is imperceptible, it was decided to modernize the approach to masking and encryption for fields such as first name, last name, and email address. The methods of pseudonymization and data fields masking: first name, last name, and email address, and the use of the FPE method in FF3-1 mode for fields such as phone number and credit card were implemented. An improved method of pseudonymization and masking of data fields is implemented in the software application.

**Keywords**: personal information, format-preserving encryption, pseudonymization, data masking.