

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

УДК 004.4



Тези доповідей

VI Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні
технології"



20-21 квітня 2023 року

Кропивницький 2023

УДК 004.4

Матеріали VI Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 20-21 квітня 2023 р. – Кропивницький: ЦНТУ, 2023. – 96 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

***За достовірність викладених фактів, цитат та інших відомостей
відповідальність несуть автори.***

© Колектив авторів, 2023
© Центральноукраїнський національний
технічний університет, 2023

УДК 004.4

Д.П. Присяжний¹, П.В. Павловський¹, В.В. Саврацький¹
dimpris@gmail.com

¹Вінницький національний технічний університет, м. Вінниця

УДОСКОНАЛЕННЯ МЕТОДУ ВИЯВЛЕННЯ ПОРУШЕННЯ КОНФІДЕНЦІЙНОСТІ ФАЙЛІВ ВИХІДНОГО КОДУ ЗА РАХУНОК ЗБІЛЬШЕННЯ ІНТЕРВАЛІВ ОБРОБКИ ТА ЗАСТОСУВАННЯ ТРАНСПОЗИЦІЇ

Захист авторських прав на програмний код як об'єкт інтелектуальної власності у сфері розробки програмного забезпечення нині є серйозною проблемою для розробників, адже існуючі механізми захисту прав на володіння об'єктом інтелектуальної власності не є достатньо адаптованими до особливостей даної предметної області. Як не існує достатньо відповідних цим особливостям юридичних механізмів, так і програмних засобів, що дозволили б чітко визначати факт порушення автентичності файлів вихідного коду програм. Оскільки програмний код з точки зору законодавства у сфері захисту авторського права розглядається як звичайний текст, то відповідно для пошуку ознак порушення автентичності файлів та відповідно доведення факту несанкціонованого копіювання із подальшим присвоєнням авторства використовуються ті ж рішення, що й для порівняння звичайних текстів.

Ще одним аспектом, що є надзвичайно важливим для алгоритмів порівняння файлів вихідного коду, є швидкість проведення перевірки. При проведенні судових чи внутрішніх службових розслідувань нині керуються набором методів та механізмів, що повинні виявляти характерний стиль автора і на основі цих доказів свідчити про автентичність чи неавтентичність файлу. Однак, дані методи орієнтуються на ручне їх виконання, тож час обробки результатів таких досліджень є непристойно великий. Таким чином, існує гостра потреба у пошуку рішення, що дозволило б проводити перевірку автентичності файлів вихідного коду програм за значно менший час аніж існуючі методи та базуватиме свої висновки на певній чіткій метриці, що не даватиме простору для ентропії.

Метою роботи є удосконалення методу виявлення порушення конфіденційності файлів вихідного коду програм.

Об'єктом дослідження є процес захисту від несанкціонованого копіювання порушення автентичності файлів вихідного коду програм.

Предметом є методи та засоби порівняння версій файлів вихідного коду програм.

У результаті дослідження виконано аналіз існуючих систем, що дозволяють проводити порівняння версій файлів та визначено, що ті алгоритми та методи, що у них використано, не здатні забезпечити необхідну об'єктивність результатів порівняння версій у контексті розслідувань випадків порушення автентичності файлу вихідного коду.

Було проаналізовано критерії до ефективності алгоритмів порівняння версій та визначено що ключовими для оцінювання успішності розробки вдосконалення методу є похибка обчислень відносно оригінального алгоритму та часова ефективність алгоритму. Визначено, що відстань Дамероу-Левенштейна є кращим рішенням при виборі метрики оцінки відмінності між послідовностями кодів файлів вихідного коду оригіналу та копії, оскільки обчислення даної метрики передбачає операції, що дозволяють зробити оцінку відмінності більш точною та об'єктивною, а також враховувати особливості предметної області, як от транспозицію у неперевірених індексах.

Виявлено, що алгоритм визначення відстані Дамероу-Левенштейна через те, що початково був розроблений для порівняння текстових послідовностей, не враховує регістр буквених символів та визначає однакові літери у різних регістрах як різні символи. У кодовій послідовності регістр літер відіграє значно меншу роль у зміні змісту коду в порівнянні із звичайним текстом. Таким чином, для досягнення більшої точності обрахунку відстані Дамероу-Левенштейна між двома послідовностями вихідного коду пропонується додати операцію виведення усіх символів обох послідовностей до одного регістру перед початком їх порівняння за алгоритмом Вагнера-Фішера.

Незалежно від того якої міри зміненою шляхом рефакторингу є послідовність копії відносно оригіналу, ці вирази завжди будуть однаковими. При цьому, саме такі вирази становлять більшу частину кодової послідовності, у той же час фактично несучи меншість впливових ресурсів на результати порівняння. Таким чином, обробка цих відрізків коду вимагає використання надлишкових ресурсів, як результат збільшуючи час обробки вхідних даних алгоритмом.

Також, з метою покращення показників швидкості проведення перевірки подібності запропонованим алгоритмом, пропонується проводити виключення усіх повторюваних та заздалегідь однакових елементів послідовностей, що обумовлено особливістю коду як текстової послідовності – незалежно від мови програмування якою написано код, у ньому завжди присутні деякі вирази, які не підлягають зміні навіть під час модифікації методами рефакторингу, оскільки це призведе до втрати кодом працездатності. Таким чином, дані

послідовності варто піддати виключенню перед початком проведення порівняння послідовностей коду, тим самим скоротивши обсяг вхідних даних, без втрати точності порівняння.

Показниками успішності реалізації запропонованого рішення є:

1. відношення результату обрахунку редакційної відстані за удосконаленою версією відстані Дамероу-Левенштейна до результатів отриманих при порівнянні тих же зразків коду за відстанню Левенштейна, як показник покращення точності визначення редакційної відстані;

2. покращення показників швидкості обробки удосконаленим алгоритмом порівняння версій коду за відстанню Дамероу-Левенштейна у порівнянні зі швидкістю обробки порівняння за відстанню Левенштейна.

Завдяки застосуванню удосконаленому алгоритму вдалося точніше визначити відстань Дамероу-Левенштейна між тестовими стрічками коду та довести, що вони є майже однаковими. Точність обчислення при цьому зросла в середньому на 33% для розглянутих випадків.

Розроблений удосконалений алгоритм передбачає скорочення довжини послідовностей коду, що перевіряються ще перед початком роботи алгоритму порівняння, отже автоматично навантаження на алгоритм у порівнянні із оригінальним рішенням буде зменшуватись. При проведенні тестування очікуваний характер поведінки зміни часової ефективності удосконаленого алгоритму відносно оригінального матиме вигляд параболи удосконаленого алгоритму буде більш пологою одночасно, ніж парабола $O(n^2)$ та параболо-видний графік $O(m*n)$ оригінального алгоритму.

Висновки. На сьогодні для проведення подібних перевірок найчастіше застосовуються алгоритми прямого порівняння, відстань Хеммінга та відстань Левенштейна. Основними недоліками цих методів є обмежена кількість операцій, які можуть виявляти ці алгоритми та врахування регістру літер як різних символів. Як альтернативу даним методам було розглянуто використання алгоритму для визначення відстані Дамероу-Левенштейна. Даний алгоритм дозволяє виявляти застосування транспозиції при виконанні рефакторингу для файлу копії, тим самим показуючи значно кращі показники точності порівняння послідовностей. Однак, і даний метод має недоліки у контексті порівняння саме файлів вихідного коду, зокрема те ж врахування регістру літер як різних символів. Рішення для усунення даного недоліку було знайдено у зведенні усіх символів обох порівнюваних файлів вихідного коду до одного регістру перед початком перевірки.

Також, було виявлено необхідність пошуку рішення для оптимізації часу виконання порівняння, адже на реальних задачах із пошуку порушення автентичності файлів вихідного коду об'єми порівнюваних даних є досить великими. Часова ефективність оригінального алгоритму для визначення відстані Дамероу-Левенштейна за Big O була визначена як $O(m*n)$, що свідчить про те, що час обробки вхідних даних алгоритмом повністю залежить від обсягу цих даних. У той же час було виявлено характерну особливість коду як текстової послідовності – він завжди матиме повторювані та невід'ємні елементи, що заздалегідь не впливають на результати порівняння, оскільки їх зміна призводить до втрати кодом працездатності. Таким чином, було запропоновано рішення додати перед початком обчислення відстані обробку файлів перевірки алгоритмом, що видалятиме із послідовностей усі завчасно однакові елементи. Для перевірки працездатності запропонованих рішень було виконано обрахунки для однакової пари послідовностей за трьома алгоритмами: оригінальним алгоритмом для відстані Левенштейна, оригінальним алгоритмом для відстані Дамероу-Левенштейна та алгоритму для відстані Дамероу-Левенштейна із додаванням запропонованих рішень.