УДК 004.56.5

МЕТОД КРИПТОГРАФІЧНОГО ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

О. В. Дмитришин, к.т.н., ст. викладач
С. В. Вузій, студент
Вінницький національний технічний університет
olexanderdm@gmail.com

За оцінками Business Software Alliance щорічні фінансові збитки через порушення авторських програмного забезпечення (ПЗ) становить $53 млрд. Тому актуальність захисту програмного зростає з кожним днем. На сьогоднішній день задача захисту ПЗ розбивається на такі напрямки:

- захист інтелектуальної власності;
- захист програмного коду;
- захист ПЗ від несанкціонованого доступу;
- захист ПЗ від несанкціонованого копіювання.

Розглянемо відомі методи захисту програмного коду. Перший з них полягає у генеруванні секретного ключа з відповідною інформацією двійкового коду під час виконання. У даному методі всі процедури дешифрування або зашифрування інших процедур, використовують їх власний код в якості ключового матеріалу, що дозволяє забезпечити цілісність даних. Якщо зловмисник намагається здійснити втручання у виконання захищеної програми, то даний метод не зможе правильно розшифрувати захищені функції, що призведе до переривання роботи програми. Даний метод має недолік, коли функція, що зашифровується, викликається декількома іншими функціями. Нехай є функції (A, B, C, і

D), як показано на рис. 1. У відповідності зі схемою, секретний ключ функції D має бути отриманий на основі коду функцій B або C, але даний метод не може визначити, який секретний ключ був використаний для зашифрування, при розшифруванні.

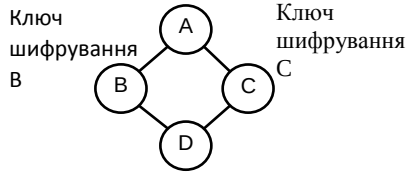

Рисунок 1 – Схема генерації ключа

Наступний метод подібний до попереднього. Даний метод зашифровує N-й блок за допомогою ключа, який був отриманий на основі (N-1)-го блоку. Якщо блок викликається більш ніж з двох попередніх блоків, то блок що викликається дублюється (рис. 2).

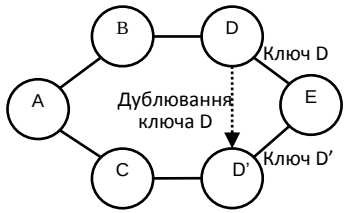

Рисунок 2 – Шифрування з дублюванням блоків

Для вирішення проблеми надлишковості було запропоновано схему на основі індексної таблиця. У цій схемі блоки B і C може розшифрувати блок D без дублювання блоку D, так як блок B і C можуть отримати ключ B з індексної таблиці, як показано на рис. 3.

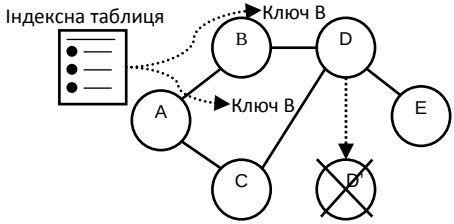

Рисунок 3 – Шифрування на основі індексної таблиці

Індексна таблиця містить необхідну для розшифрування інформацію (приклад індексної таблиці табл. 1).

Таблиця 1 – Приклад індексної таблиці.

Адреса	Розмір блоку	Кількість викликів	Прапор
0x0040103E	19	1	0
0x00401051	9	2	1
0x0040105A	6	2	1
0x00401060	12	1	0

Даний метод має два недоліки. Перший полягає у тому, що не можливо забезпечити повністю цілісність, оскільки функції, які викликаються з декількох функцій зашифровуються на основі псевдовипадкового числа. Адже таке псевдовипадкове число можна підібрати і отримати доступ до даної функції, та змінити її. Другий недолік витікає з першого і полягає він у тому що даний метод не може забезпечити достатній рівень конфіденційності.

Запропонуємо метод який забезпечить більш надійну за попередні методи конфіденційність. Суть методу полягає в тому що для шифрування буде використовуватись 128-бітний ключ, на основі шифру AES. За формування ключа відповідатиме функція – $K_i = f(k, ID_{function})$. Спочатку потрібно про індексувати всі функції щоб отримати $ID_{function}$. Таким чином будуть сформовані індивідуальні ключі для кожної функції програми, що значно підвищує її конфіденційність.

Висновок в дані роботі було досліджено існуючі методи блокового шифрування та запропонований вдосконалений метод, який враховує недоліки попередніх. В подальшому планується дослідження ефективності запропонованого методу та його реалізація.