УДК 621.391.7

А. В. Приймак

ПІДВИЩЕННЯ КРИПТОСТІЙКОСТІ СИМЕТРИЧНОГО ШИФРУ ШЛЯХОМ ВИКОРИСТАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ

Вінницький національний технічний університет

Анотація

В роботі розглянуто та досліджено проблему створення слабких ключів для криптографічного алгоритму IDEA, а також запропоновано підвищення його криптостійкості шляхом використання генетичного алгоритму. Оскільки ці ключі використовуються для шифрування та дешифрування, то можна легко спрогнозувати шифр тексту, що відповідає відкритому тексту. Для застосування генетичного підходу, який є відомим методом оптимізації, для слабких ключів було отримано нове рішення для перетворення слабких ключів на більш сильні. Можливості створення слабкого ключа в IDEA є рідкісними, але, якщо вони все ж генеруються, то це може призвести до швидкого взлому шифротексту та втрати інформації.

Ключові слова: криптографія, ключ, генетичний алгоритм, IDEA, криптостійкість.

Abstract

The problem of creation of weak keys for the IDEA cryptographic algorithm is considered and investigated in this article, and the way of increasing its cryptostability is suggested by using the genetic algorithm. Since these keys are used for encryption and decryption, it is easy to predict the encrypted part of text that is corresponding to the source text. For the use of the genetic approach, which is a well-known optimization method, for the weak keys, a new solution was obtained to transform the weak keys into more powerful ones. The ability to create a weak key in IDEA is rare, but if they are still generated, it can lead to a quick hacking of ciphertext and loss of information.

Keywords: cryptography, key, genetic algorithm, IDEA, cryptostability.

Вступ

Криптографія призначена для передачі захищених даних через незахищену мережу в зашифрованому варіанті, щоб лише один із користувачів, якому призначена ця інформація, міг проаналізувати його. Зв'язок через повідомлення, електронні листи або різні інші режими вимагає високої безпеки, щоб зберігати конфіденційність вмісту. У даній роботі розглядається недолік IDEA - створення слабких ключів. Оскільки ці ключі використовуються для шифрування та дешифрування, то можна легко спрогнозувати шифротекст, що відповідає відкритому тексту. Для застосування генетичного підходу, який є відомим методом оптимізації, для слабких ключів було отримано нове рішення для перетворення слабких ключів на більш сильні. Можливості створення слабкого ключа в IDEA є рідкісними, але, якщо вони створюються, то це може призвести до втрати інформації. Отже, на сьогоднішній день є дуже важливо вжити заходи для захисту ключових елементів та забезпечення конфіденційності інформації, шляхом підвищення криптостійкості алгоритму IDEA. У криптографії слабкий ключ є ключем, який, використовуючи спеціальний шифр, змушує шифр вести себе якимось небажаним чином [1]. Слабкі ключі зазвичай являють собою дуже невелику частину загального простору ключів, що зазвичай означає, що, якщо хтось генерує випадковий ключ для шифрування повідомлення, слабкі ключі навряд чи викличуть проблему для безпеки шифру, оскільки шанс генерації даного ключа для певного алгоритму шифрування досить малий. Проте, вважається, що шифр не повинен мати слабких ключів, для того, щоб його можна було вважати стійким шифром. Також вважається, що шифр без слабких ключів має плоский або лінійний ключовий простір

Метою роботи є підвищення криптостійкості симетричного алгоритму IDEA шляхом використання генетичного алгоритму.

Результати дослідження

Генетичні алгоритми (ГА) – це клас алгоритмів оптимізації. ГА має на меті вирішення завдань шляхом моделювання спрощеної версії генетичних процесів. Є багато проблем, для яких підхід ГА є корисним.

Тому в цій роботі досліджується використання ГА в криптографічному алгоритмі IDEA для підвищення його криптостійкості. Як традиційний криптоаналіз, так і ГА-методи реалізовані в програмному забезпеченні. Результати потім порівнюються, використовуючи показники пройденого часу та відсоток успішних дешифрувань. Встановлюється визначення кожного розглянутого шифру стосовно обґрунтованості підходів, що базуються на ГА [2].

Основна ідея ГА полягає в тому, щоб моделювати процес природного відбору, де застосовуються генетичні оператори для покращення генерації. Поетапно генетичний алгоритм виглядає наступним чином:

Оператор відбору: призначення цього оператора полягає у виборі кращих батьків, щоб передавати кращі характеристики до наступного покоління. Переваги кожного окремого покоління в певному поколінні залежать від його придатності, яка може бути розрахована об'єктивною функцією або суб'єктивним судженням.

Схрещення: за допомогою оператора виділення вибираються два найкращі об’єкти з множини об’єктів, а також вибирається випадкова точка схрещення. Біти міняються місцями в рядах бітів вибраних об’єктів, враховуючи випадкову точку схрещення [3].

Слабкість IDEA полягає в можливій генерації слабкого ключа, що використовується для шифрування та дешифрування. Цей ключ може бути в подальшому виявленим при атаці на основі підібраного відкритого тексту.

Всього для алгоритму IDEA було виявлено 3 великі класи слабких ключів [4]:

Перша категорія слабких ключів включає ключі, які відповідають за лінійний коефіцієнт (наприклад, лінійний зв'язок між певними вхідними та вихідними бітами, які мають певну вірогідність).

Друга категорія слабких ключів включає в себе ключі, для яких зміна деяких біт вводу робить зміну і вихідних біт, яка може бути в подальшому ідентифікована з певною ймовірністю.

Третя категорія слабких ключів включає в себе ключі, які передбачувані, якщо відомо, що деякі біти зашифрованого тексту відповідають бітам відкритого тексту.

Дані класифікації слабких ключів не можуть використовуватися для шифрування і дешифрування інформації, а отже, це вимагає подальшого вдосконалення даного алгоритму. Запропонований метод вирішує це завдання, застосовуючи генетичний алгоритм до слабкого ключа. Це призводить до створення сильного ключа із слабкого, який уже може бути використаний для цілей шифрування і дешифрування [5].

Покращений алгоритм генерації ключа для шифрування та дешифрування складається з 7-ми головних кроків:

Крок 1. Генерується один випадковий ключ (K).

Крок 2. Щоб перевірити, чи ключ (K) - сильний або слабкий, виконується перевірка на основі категорій слабких ключів, які були описані раніше. Кожна слабка категорія ключів має заздалегідь визначений формат. Якщо ключ виявиться слабким, потрібно перейти до кроку 3. Інакше можна переходити до кроку 7.

Крок 3. Необхідно розділити ключ (K) на два підключі (K_1 і K_2), кожен з яких складається з 64 біт. Вони представляють два вихідних блоки, які братимуть участь у генетичному процесі для виробництва покращених дочірних ключів.

Крок 4. Випадково генерується точка схрещення (від 0 до 63 біт) і виконується сам процес схрещення між K_1 і K_2 для створення двох дочірніх підключів (C_1 і C_2).

Крок 5. Після операції схрещення, два дочірні підключі (C_1 і C_2) об'єднуються, щоб створити 128-бітний ключ (K').

Крок 6. Після створення ключа виконується операція мутації. Випадково генерується 10 чисел в межах від 0 до 127. Ці числа є номерами бітів, які будуть замінені під час мутації. Після процесу мутації новостворений ключ позначається як K.

Крок 7. Тепер можна використовувати створений сильний ключ для шифрування та дешифрування у алгоритмі IDEA.

Процес генетичної генерації симетричного ключа для шифрування та дешифрування зображений на рисунку 1.

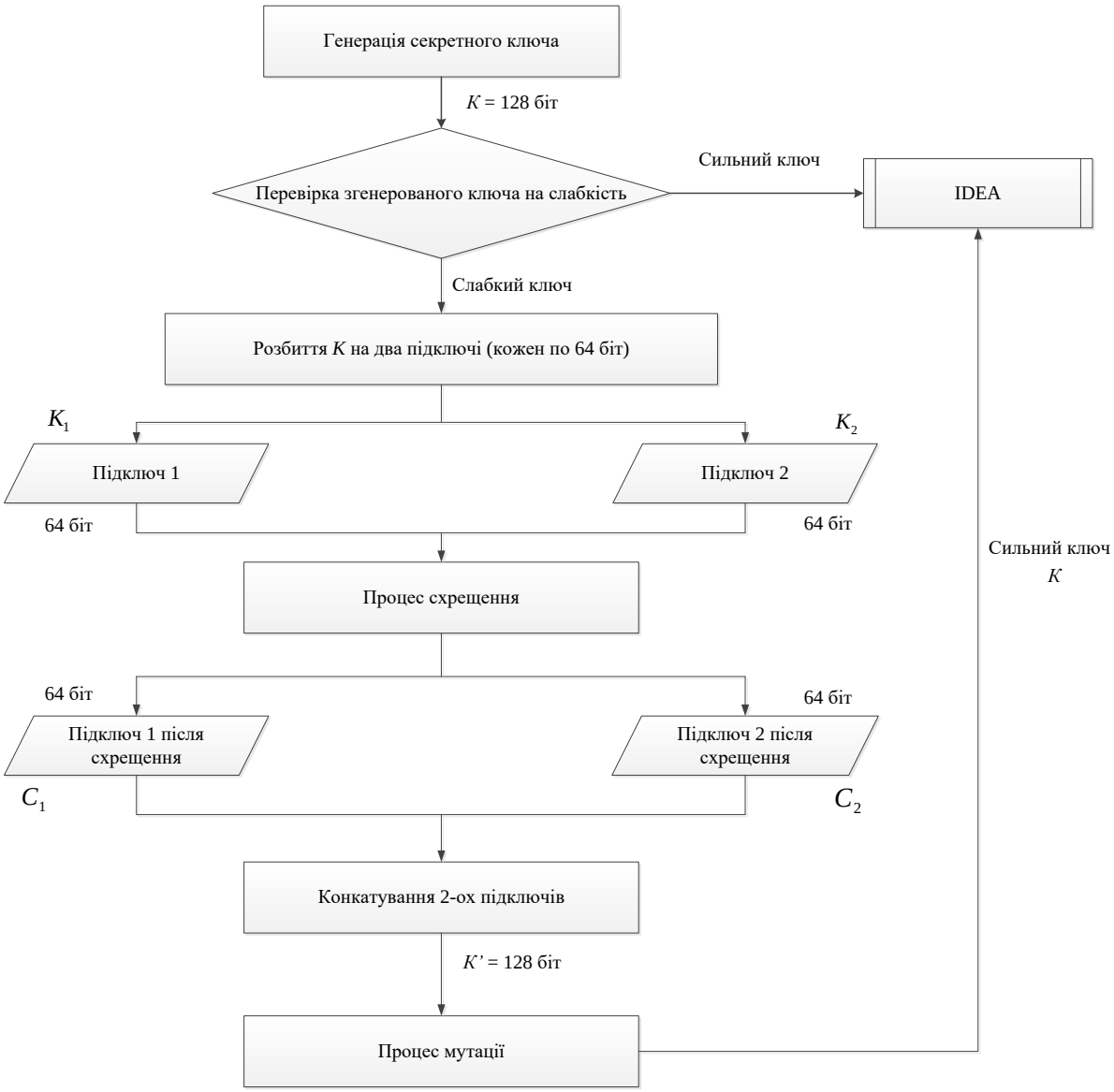

Рисунок 1 – Запропонований алгоритм генерації секретного ключа

Сильний ключ, отриманий в результаті виконання алгоритму, що описаний вище, вже може в подальшому використовуватись для цілей шифрування в IDEA. Оскільки IDEA є симетричним шифром, то отриманий сильний ключ буде використовуватись і для дешифрування шифротексту.

Варто зазначити, що слабкий ключ, після проходження оптимізації, гарантовано стає сильним ключем. Тому дана процедура оптимізації не потребує зайвого порівняння оптимізованого ключа з класами слабких ключів. Це є важливою перевагою запропонованого методу оптимізації, оскільки у випадку простої заміни байтів початкового ключа на сильні, є вирогідність створення знову ж слабкого ключа, тому для виконання такої функції необхідне застосування повторного порівняння зміненого та оригінального ключа, що викличе циклічність, а отже більші затрати часу на оптимізацію ключа та навантаження на роботу процесора.

Висновки

У даній роботі було запропоновано підвищення криптостійкості симетричного шифру IDEA за рахунок використання генетичного алгоритму для оптимізації слабких ключів. Також було детально описано запропонований генетичний алгоритм оптимізації слабких ключів. Покроково описана структура запропонованого алгоритму.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гулак Г. М. Основи криптографічного захисту інформації / Г.М. Гулак, В.А. Мухачов, В.О. Хорошко, Ю.Є. Яремчук – Вінниця, 2011 – 199 с.

2. Азаров О. Д.. Комп'ютерна криптографія / Азаров О. Д., Хорошко В. О., Шєлєст М. Є., Мухачьов В. А., Яремчук Ю. Є. - НАУ, 2003 – 14 с.

3. Ю. Є. Яремчук. Сучасний захист інформації / Ю.Є. Яремчук, А. П. Бондарчук, С. Я. Довбня, Ю. І. Хлапонін – Вінниця, 2013.

4. IDEA [Electronic Resource]. – Mode of access : URL : https://uk.wikipedia.org/wiki/IDEA- Назва з екрану.

5. Weak key [Electronic Resource]. – Mode of access : URL : https://en.wikipedia.org/wiki/Weak_key. - Назва з екрану.

Приймак Андрій Васильович — магістр, Вінницький національний технічний університет, Вінниця, e-mail: andrii.pryimak@live.com.

Науковий керівник: ***Яремчук Юрій Євгенович*** — доктор технічних наук, професор, факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, Вінниця.

Pryimak Andrii Vasyliovych — master degree, Vinnitsa National Technical University, Vinnitsa, e-mail: andrii.pryimak@live.com.

Supervisor: ***Yaremchuk Yuriy E***. — Ph. D., professor, management and security of information Systems department; Vinnitsa.