**УДК 004.056.55**

*Гриник Р.О.*

*Львівський державний університет безпеки життєдіяльності, м. Львів*

ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ЗАДАЧ КРИПТОАНАЛІЗУ

Описано суть генетичного алгоритму та принципи роботи основних генетичних операторів: кросинговер, мутація, селекція. На основі аналізу результатів останніх досліджень обґрунтовано доцільність застосування генетичного алгоритму в задачах криптографічного аналізу симетричних та асиметричних алгоритмів шифрування. Наведено приклади криптографічного аналізу, за допомогою генетичного алгоритму, алгоритму RSA та потокових шифрів типу шифру Віженера.

Ключові слова: генетичний алгоритм, селекція, кросинговер, мутація, криптографічний аналіз

Описаны суть генетического алгоритма и принципы работы основных генетических операторов: кроссинговер, мутация, селекция. На основе анализа результатов последних исследований обоснована целесообразность применения генетического алгоритма в задачах криптографического анализа симметричных и асимметричных алгоритмов шифрования. Приведены примеры криптографического анализа, с помощью генетического алгоритма, алгоритма RSA и потоковых шифров типа шифра Виженера.

Ключевые слова: Генетический алгоритм, селекция, кроссинговер, мутация, криптографический анализ

Describe the nature of the genetic algorithm and principles of basic genetic operators: crossing over, mutation breeding. Based on the analysis of the results of recent studies proved the feasibility of genetic algorithm in problems of cryptanalysis of symmetric and asymmetric encryption algorithms. Examples of cryptographic analysis, using a genetic algorithm, RSA algorithm type and stream cipher Vigenère cipher.

Keywords: Genetic algorithm selection, crossing over, mutation, cryptographic analysis

***Вступ***

Застосування криптографічних методів для захисту інформації набуває все більшої популярності і в наш час є одним з найбільш важливих. На сьогоднішньому етапі розвитку обчислювальної техніки до алгоритмів шифрування виставляються все вищі вимоги, оскільки із зростанням можливостей обчислювальної техніки зростають можливості криптоаналітиків. На сьогоднішній день все ширшого застосування в криптоаналізі набувають такі еволюційні алгоритми як генетичний алгоритм, алгоритм імітації відпалу металу та алгоритм роєвого інтелекту.

Основна ідея генетичних алгоритмів – відтворення випадковості природного відбору, де популяція особин адаптується до середовища існування через процес природного відбору. Це означає, що виживання і відновлення особини обумовлюється усуненням небажаних ознак. Для вдосконалення нащадків генетичний алгоритм використовує ітераційне застосування набору випадкових операторів - таких як мутація, кросинговер, селекція.

***Принцип роботи генетичного алгоритму***.

Генетичний алгоритм, для формування нової популяції з кращою придатністю,використовує нижче описані оператори.

*1. Кросинговер (схрещення)*

Кросинговер – це генетичний оператор, який призначений для утворення нової хромосоми-нащадка в результаті схрещення двох хромосом-батьків. Кросинговер буває двох видів:одноточковий та двоточковий. В одноточковому кросинговері випадковим чином визначається точка всередині хромосоми (точка розриву), в якій обидві хромосоми діляться на дві частини і обмінюються ними. У двоточковому кросинговері хромосоми розглядаються як цикли, які формуються з'єднанням кінців лінійної хромосоми. Для заміни сегменту одного циклу сегментом іншого необхідно обрати дві точки розриву. У цьому поданні одноточковий кросинговер може бути розглянутий як кросинговер з двома точками, але з однією точкою розрізу, зафіксованою на початку рядка[1].

*2. Мутація*

Мутація – це генетичний оператор, який змінює одне, або декілька бітових значень у хромосомі. Даний оператор застосовується до хромосоми-нащадка для гарантування знаходження усього простору можливих вирішень задачі. Мутація виконується досить рідко, оскільки значна кількість змінених бітів у хромосомі-нащадку призведе до повної її відмінності від хромосом-батьків. Вважається, що коефіцієнт мутації не повинен перевищувати 10% [1].

*3. Селекція(відбір)*

Селекція – це стадія,на якій генетичний алгоритм відбирає хромосоми з існуючої популяції для рекомбінації на основі значення фітнес-функції. Хромосома з більшим значенням фітнес-функції буде вважатися кращою.

Різновиди селекції:

* Селекція «колесо рулетки». Процедура селекції повторюється до ти, доки не буде достатньо відібрано осіб для нової популяції [2].
* Селекція-турнір. Відбір у нову популяцію базується на виборі кращого індивіда з випадково вибраної підгрупи[2].
* Селекція усічення. Передбачає вибір кращої половини, третини або іншої частини осіб у нову популяцію[2].
* Відбір витісненням. У даному відборі вибір особи в нову популяцію залежить не тільки від величини її придатності, але і від того, чи є вже в сформованій популяції особина з аналогічним хромосомним набором [1].
* Елітарний відбір. Створюється проміжна популяція, яка включає в себе як батьків, так і їхніх нащадків. Члени цієї популяції оцінюються, а після з них вибираються N найкращих (придатних) осіб, які увійдуть в наступне покоління [1].

Розмір популяції залишається незмінним протягом всього періоду роботи генетичного алгоритму. Визначення розміру популяції є одним з основних факторів, які вплинуть на якість вирішення задачі. Занадто малий розмір популяції збільшує ризик передчасного сходження до локальних мінімумів.

Функція придатності відіграє дуже важливу роль у керуванні генетичним алгоритмом. Хороша функція придатності допоможе генетичним алгоритмам досліджувати пошукову систему більш ефективно. Погана функція придатності може легко ввести генетичний алгоритм у пастку.

***Застосування генетичних алгоритмів в криптоаналізі***

На сьогоднішній день генетичні алгоритми використовують при криптоаналізі потокових шифрів [4], генерації ключів асиметричних алгоритмів шифрування [2], здійсненні криптоаналізу симетричних шифрів.

Криптоаналіз потокових шифрів здійснюється в два етапи: на першому етапі визначається довжина ключа, а на другому – сам ключ шифрування. Для знаходження довжини секретного ключа спочатку генетичний алгоритм запускають у припущенні, що довжина секретного ключа дорівнює деякому *l*, і після завершення роботи алгоритму запам'ятовують «рекордне» (тобто найменше) значення *c*[*l*]фітнес-функції за весь час роботи алгоритму. Потім параметр *l* збільшується на *1, 2, 3* і т.д., щоразу заново запускаючи генетичний алгоритм, а після його завершення обчислюють «рекордне» значення *c*[*l+1*]*, c*[*l+2*]*, c*[*l+3*] і т.д. фітнес-функції для ключів довжини (*l+1*)*,* (*l+2*)*,* (*l+3*)і т.д. Найменше серед знайдених «рекордних» значень фітнес-функції досягається саме при спів падінні довжини, яку ми перевіряємо,і довжини секретного ключа. Нехай *n* – довжина секретного ключа. Якщо числа *n* і *l* збігаються, то в результаті роботи генетичного алгоритму буде знайдений сам таємний ключ, або близький до нього ключ, а фітнес-функція, при використанні цього ключа, прийме відносно невелике значення. Повторна робота генетичного алгоритму над зашифрованим текстом, але з уже відомою довжиною секретного ключа,дозволить знайти сам секретний ключ або близький до нього ключ[4]. Як фітнес-функцію рекомендується використовувати функцію Якобсена про розподіл частот біграм у відкритому тексті[5].

Поряд з класичними симетричними алгоритмами шифрування порівняно молодою областю є асиметрична криптографія, яка включає криптосистеми з відкритим ключем [3]. Представником її є алгоритм RSA, крипостійкість якого визначається трудомісткістю факторизації великих чисел [5]. Задача криптоаналітика зводиться до розкладання на множники великого числа. Генетичний алгоритм вирішує цю задачу таким чином: задається число в двійковій формі; генерується початкова популяція, яка складається з двох частин: перша частина відповідає першому множнику, друга частина – другому; виконуються генетичні операції з популяціями, формується нова популяція та підраховується значення фітнес-функції. Як фітнес-функція використовується множення хромосом з ідентичними номерами із кожної частини [6].

При криптоаналізі асиметричних алгоритмів шифрування актуальною також є задача знаходження простого дільника заданого числа[6]. Для знаходження простого дільника числа необхідно задати число та перевести його в двійкову форму; згенерувати випадковим чином початкову популяцію;обчислити фітнес-функцію;перевірити число на простоту за допомогою одного з відомих алгоритмів. Якщо число не просте, то за допомогою генетичних операцій формується нова популяція та обчислюється фітнес-функція. Процес продовжується доти, доки не отримається просте число. Як фітнес-функція використовується ділення заданого числа на хромосому (ділення проводиться в десятковій формі). Результатом є залишок від ділення.

***Висновок***

Проведено дослідження можливостей застосування генетичного алгоритму в криптоаналізі. Встановлено особливості використання генетичного алгоритму для здійснення криптоаналізу потокових шифрів та асиметричних алгоритмів шифрування. Наведено рекомендації щодо формування фітнес-функцій в конкретному випадку.

**Література:**

1. Панченко Т.В. Генетичні алгоритми// Издательский дом «Астраханский университет» 2007.
2. Swati Mishra, Siddharth Bali. Public Key Cryptography Using Genetic Algorithm// International Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878, Volume-2, Issue-2, May 2013.
3. Основы криптографии: Учебное пособие / Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. – М. : Гелиос АРВ, 2001. –
4. Морозенко В. В. Генетический алгоритм для криптоанализа шифра Виженера / Морозенко В. В., Елисеев Г. О. // Весник Пермского университета – 2010. –
5. Чернышев Ю. О. Применение биоинспирированных методов оптимизации для реализации криптоанализа классических и блочных криптосистем / Чернышев Ю. О., Сергеев А. С., Дубров Е. О. // Теоретические и прикладные вопросы современных информационных технологий : Материалы 11 Всероссийской научно-технической конференции. – Улан-Удэ : Изд-во ВСГТУ, 2012. – С. 121–131.
6. Чернышев Ю.О. Применение биоинспирированных алгоритмов оптимизации для реализации криптоанализа классических и асимметричных криптосистем / Чернышев Ю.О., Сергеев А.С., Дубров Е.О. // Информатика: проблемы, методология, технологии : Материалы XIV Междунар. науч. – метод.конф. – Воронеж : Изд. дом ВГУ, 2014. – С. 206-210.