ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 681.3.001.63

Р.О. Гриник, О.І. Полотай, *канд. техн. наук*
(Львівський державний університет безпеки життєдіяльності)

ЗАСТОСУВАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ РОЗКРИТТЯ РАНЦЕВОЇ КРИПТОСИСТЕМИ МЕРКЛЕ-ХЕЛМАНА

Розроблено модель та програмне забезпечення для розкриття ранцевої криптосистеми Меркле-Хеллмана на основі генетичного алгоритму. Наведено основні принципи роботи криптосистеми та генетичного алгоритму. Показано основні етапи роботи генетичного алгоритму та наведено методи, які використовувались при побудові алгоритму для криптоаналізу криптосистеми Меркле-Хеллмана. Наведено експериментальні результати, які були отримані при криптоаналізі криптосистеми з різною довжиною ключа за допомогою генетичного алгоритму з різними методами відбору батьківської пари для створення нової хромосоми, а також для розкриття криптосистеми традиційним способом.

Ключові слова: криптоаналіз, криптографія, генетичний алгоритм, криптосистема Меркле-Хеллмана.

Р.О. Грынык, О.И. Полотай

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РАСКРЫТИЯ РАНЦЕВОЙ КРИПТОСИСТЕМЫ МЕРКЛЕ-ХЕЛЛМАНА

Разработана модель и программное обеспечение для раскрытия ранцевой криптосистемы Меркле-Хеллмана на основе генетического алгоритма. Приведены основные принципы работы криптосистемы и генетического алгоритма. Показаны основные этапы работы генетического алгоритма и приведены методы, использованные при построении алгоритма для криптоанализа криптосистемы Меркле-Хеллмана. Приведены экспериментальные результаты, полученные при криптоанализе криптосистемы с разной длиной ключа с помощью генетического алгоритма с различными методами отбора родительской пары для создания новой хромосомы, а также для раскрытия криптосистемы традиционным способом.

Ключевые слова: криптоанализ, криптография, генетический алгоритм, криптосистема Меркла-Хеллмана.

R.O. Grynyk, O.I. Polotaj

USING GENETIC ALGORITHM FOR CRACKING MERKLE-HELLMAN CRYPTOSYSTEM

It is developed the model and software for disclosure knapsack cryptosystem of Merkle-Hellman based on genetic algorithm. It is shown the basic principles of the cryptosystem and genetic algorithm. Is shown the basic steps of the genetic algorithm and also the methods that are used in the construction an algorithm for cryptanalysis the cryptosystem of Merkle-Hellman. There are experimental results, which have been obtained during the cryptosystem cryptanalysis using different length of key, based on a genetic algorithm with different methods of selection the parental pair to create new chromosomes, as well as and for disclosure the cryptosystem in traditional way.

Key words: cryptanalysis, cryptography, genetic algorithm, cryptosystem of Merkle-Hellman.

Постановка проблеми

В сучасних умовах швидкого розвитку кіберпростору, все більшого значення набувають проблеми криптографічного захисту інформації, що в свою чергу викликає необхідність дослідження уразливостей криптографічних систем для подальшого їх покращення. Для цього однією з задач виступає дослідження стійкості ранцевої криптосистеми Меркла-Хеллмана до атак, побудованих на базі генетичного алгоритму.

Аналіз останніх публікацій та досліджень. Проблемами дослідження застосування еволюційних алгоритмів для здійснення розкриття криптографічних систем займаються ряд світових та вітчизняних вчених, серед яких Єлісєєв Г.О., Чернишов Ю.О., Панченко Т.В., Явурек М., Тхада В.

Виклад основного матеріалу. Ранцева криптосистема Меркле-Хеллмана є однією з перших асиметричних криптосистем і була заснована у 1978 році Мартіном Хеллманом та Ральфом Меркле на основі «задачі про рюкзак». «Задача про рюкзак» відноситься до NP-задач Річарда Карпа і може бути сформульована таким чином: існує ранець певного об'єму V, а також існує набір об'єктів $B = \{b_1, b_2, ..., b_n\}$, кожен з яких також має власну вагу та користь (вартість). Необхідно обрати множину об'єктів у такий спосіб, аби максимізувати загальну вартість (користь) множини об'єктів поміщених у рюкзак. Тобто задача зводиться до того, щоб знайти бінарний вектор $x = \{x_1, x_2, ..., x_n\}$, щоб

$$M = \sum_{i=1}^{n} x_i b_i \tag{1}$$

де $x_i \in \{0,1\}$, $i = 1, 2, ..., n$. Це означає, що якщо $x_i = 1$, то предмет b_i необхідно помістити в ранець, в іншому випадку предмет в ранець не поміщаємо.

Для розробки алгоритму шифрування Ральф Меркле використав не довільну послідовність b_i, а суперзбільшену послідовність, таку щоб

$$b_{i+1} > \sum_{i=1}^{n} b_i , \tag{2}$$

тобто таку послідовність в котрій вага кожного наступного об'єкта перевищує сумарну вагу всіх попередніх об'єктів. Неважко переконатися, що для такого набору чисел рішення задачі є тривіальним [1]. Для того, щоб позбутися тривіальності і необхідно ввести «закритий ключ», який складається з двох чисел *q* та *r*, які задовольняють такі умови:

$$1.\ q > \sum_{i=1}^{n} b_i;$$

$$2.\ r \in [1, q);$$

$$3.\ НСД\ (q, r) = 1;$$

Для шифрування інформації кожен символ повідомлення необхідно представити як бінарний вектор $m = \{m_1, m_2, ..., m_n\}$, однакової довжини, тобто як послідовність 0 і 1. Сформувати суперпослідовність $b = \{b_1, b_2, ..., b_n\}$, вибрати пару чисел q та r та сформувати відкритий ключ $k = \{k_1, k_2, ..., k_n\}$, за допомогою рівняння:

$$k_i = r b_i \bmod q \tag{3}$$

Зашифрувати повідомлення знайшовши суму всіх добутків x_i та k_i відповідно, тобто:

$$C = \sum_{i=1}^{n} k_i m_i \tag{4}$$

Для того щоб розшифрувати криптограму, одержувач повинен визначити біти повідомлення m_i, які задовольняли б формулу 4. Для цього необхідно знайти таке ціле число r', яке є мультиплікативним оберненим до числа r за модулем q, тобто r' повинно задовольняти рівняння

$$r' \times r (\mathrm{mod}\, q) = 1, \quad (5)$$

оскільки r взаємно просте з q, то r' можна знайти з допомогою розширеного алгоритму Евкліда. Після того, як знайдено r', необхідно виконати таке обчислення:

$$C' = C \times r' (\mathrm{mod}\, q) \quad (6)$$

Після цього отримане число C' необхідно розкласти на доданки таким чином: спочатку вибирати найбільший елемент з множини $b = \{b_1, b_2, ..., b_n\}$, який менший, ніж C', і обчислити їх різницю. Далі необхідно вибирати наступний найбільший елемент, який менший, ніж отримана різниця, і повторювати ці дії, доки різниця стане дорівнювати нулю. Елементи, які були обрані з множини $b = \{b_1, b_2, ..., b_n\}$, будуть відповідати 1 в двійковому записі вихідного тексту, а які не були обрані будуть відповідати 0.

Побудова моделі

На сьогодні генетичні алгоритми використовують при криптоаналізі потокових шифрів, генерації ключів асиметричних алгоритмів шифрування, здійсненні криптоаналізу симетричних та асиметричних шифрів [2]. 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. Загалом роботу цього алгоритму видно на рисунку 1.

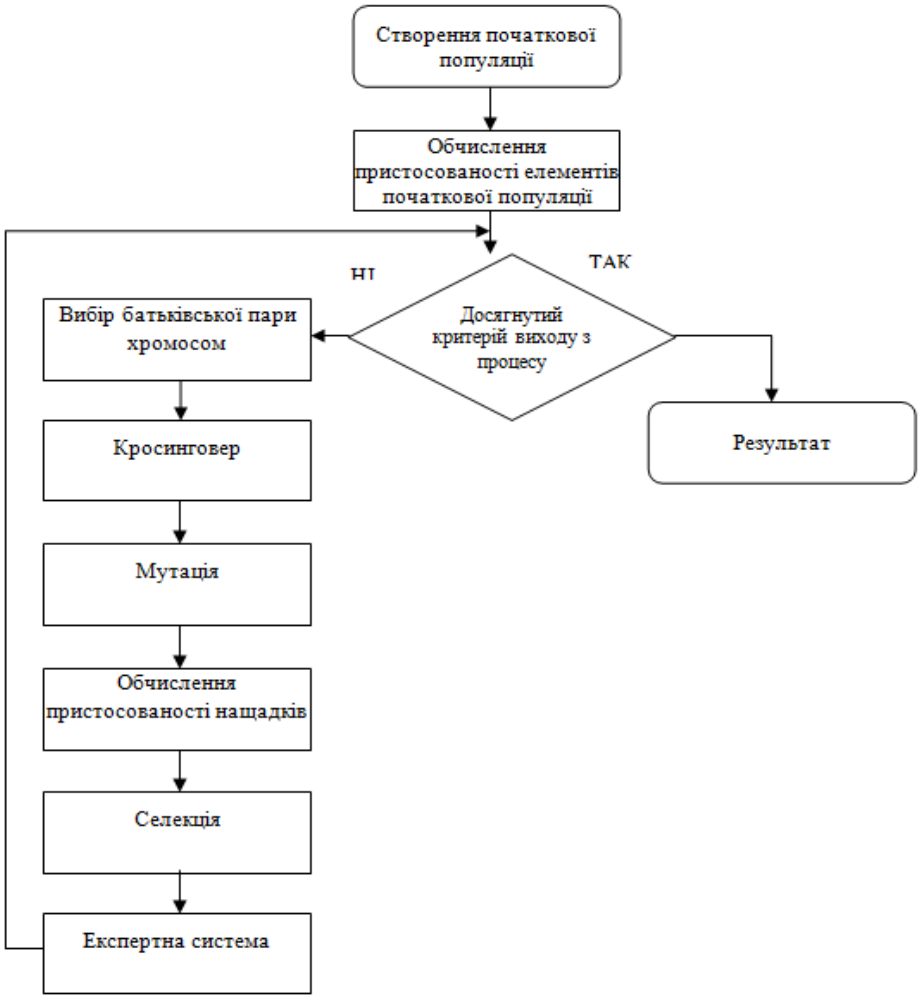

Рисунок 1 *– Схема роботи генетичного алгоритму*

При вирішенні задачі криптоаналізу з допомогою генетичного алгоритму перш за все необхідно вирішити чотири основних завдання, а саме:

1. Визначити спосіб представлення хромосоми
2. Вибрати спосіб здійснення селекції
3. Визначити цільову функцію
4. Вибрати оператор рекомбінації
5. Визначити, за яким принципом будуть відбиратись хромосоми-батьки для створення нової особини

Структура хромосом. Спосіб представлення хромосоми досить сильно впливає на ефективність генетичного пошуку, оскільки кодування спільно з оператором рекомбінації визначає ефективність обміну інформацією між популяціями [3]. Для цієї задачі найбільш оптимальний варіант кодування той при якому довжина стрічки дорівнює складності вхідної множини і являє собою бітову стрічку, котра є одним з можливих варіантів розв'язку задачі. У нашій роботі для представлення можливого варіанта розв'язку задачі ми використовували бінарні хромосоми, кожен ген якої може набувати цілих значень в інтервалі [0, 1].

Створення початкової популяції. Для створення початкової популяції необхідно визначити довжину відкритого ключа n та згенерувати випадковим чином заздалегідь задану кількість бінарних векторів (хромосом) $x = \{x_1, x_2, ..., x_n\}$, довжина кожного з яких буде дорівнювати довжині відкритого ключа.

Цільова функція. Цільову функцію необхідно побудувати таким чином, щоб з наближенням знайденого рішення до реального, значення фітнес функції прямувало до 0. Припустимо, що ми маємо вектор $k = \{k_1, k_2, ..., k_n\}$, довжини n, який являється відкритим ключем для нашого шифру, а також нам відома сама криптограма *C*. Тоді ми можемо оцінити наближеність вихідного повідомлення $b = \{b_1, b_2, ..., b_n\}$ до поточного рішення задачі $x = \{x_1, x_2, ..., x_n\}$, як різницю між реальним шифротекстом C і шифротекстом, отриманим з вектора $x = \{x_1, x_2, ..., x_n\}$:

$$f(x) = \left| C - \sum_{i=1}^{n} x_i k_i \right|. \tag{7}$$

При цьому, якщо значення функції дорівнює нулю, то криптограми збігаються, що означає, що вектори $x = \{x_1, x_2, ..., x_n\}$ і $b = \{b_1, b_2, ..., b_n\}$ збігаються. 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 призвести до неможливості вирішення поставленої задачі.

Принцип відбору хромосом-батьків. Існує кілька підходів для вибору батьківської пари хромосом. У цій роботі для порівняння ми використовували випадковий вибір батьківської пари з усієї популяції, та метод рулетки. Суть турнірного методу рулетки полягає в тому, що особини відбираються за допомогою *N* «запусків» рулетки, де *N* - розмір популяції. Колесо рулетки містить по одному сектору для кожного члена популяції. Розмір *i-го* сектора пропорційний ймовірності попадання в нову популяцію $P(i)$, що обчислюється за формулою:

$$P(i) = \frac{f(i)}{\sum_{i=1}^{N} f(i)} \tag{8}$$

де $f(i)$ – придатність *i-ої* особини. Очікуване число копій *i-ої* хромосоми після застосування оператора рулетки визначається за формулою:

$$N_i = P(i)N \tag{9}$$

При такому відборі члени популяції з більш високою пристосованістю будуть частіше вибиратися, ніж особини з нижчою пристосованістю [4].

Оператор рекомбінації. Оскільки у цій задачі неможливо визначити взаємозв'язок біт в хромосомах батьків, що являється головною ознакою для використання одноточкового чи багатоточкового кросинговеру, то у роботі використовувався рівномірний кросинговер [3]. Його суть полягає в тому, що рівномірний кросинговер потребує визначати значення кожного біта хромосоми-нащадка шляхом випадкового вибору – від кого з хромосом-батьків нащадок отримає даний біт.

Мутація. Цей 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% [5].

Селекція. Селекція – це стадія, на якій генетичний алгоритм відбирає хромосоми з наявної популяції для рекомбінації на основі значення фітнес-функції. Для вирішення цієї задачі використовувався елітарний відбір. Суть цього методу селекції полягає в тому, що створюється проміжна популяція, яка включає в себе як батьків, так і їхніх нащадків. Члени цієї популяції оцінюються, а потім з них обираються N найкращих (придатних) осіб, які увійдуть у наступне покоління [4].

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.

Загальна схема алгоритму

Алгоритм поділяється на сім основних кроків:

1. Створення популяції заданої розмірності
2. Обчислення пристосованості кожної особини популяції, якщо існує особина з значення фітнес функції нуль, то перейти до пункту 7.
3. Застосувати функцію рекомбінації
4. Обчислити пристосованість новоутвореної популяції
5. Застосувати оператор мутації
6. За допомогою селекції відібрати хромосоми у нову популяцію
7. Обчислити пристосованість новоутвореної популяції, якщо з значення фітнес функції жодна особина не дорівнює нулю, перейти до пункту 3.
8. Завершити виконання програми.

Експериментальні результати

Для проведення експериментальних обчислень, за допомогою мови програмування C# у середовищі Visual Studio 2015, було розроблено програмне забезпечення, яке дає змогу перевірити стійкість криптосистеми до атак, що побудовані на основі генетичного алгоритму. Для цього було змодельовано атаки на криптосистему Меркле-Хеллмана з різною довжиною ключа, а також атаки «грубою силою» (методом перебору). При реалізації цих атак на основі генетичного алгоритму використовувались два різних способи вибору батьківських хромосом, що дали різні результати, які наведено в табл.1-2. Під час проведення експериментальних обчислень, повідомлення та ключ генерувались випадковим чином, повідомлення шифрувалось та виконувалась атака. З кожним ключем, ця процедура виконувалась 100 разів, після чого обчислювалось середнє значення кількості обчислень фітнес функцій для знаходження розв'язку а також відсоток успішних атак. В таблиці 1 наведені результати за яких розмір популяції – 50 осіб, а коефіцієнт мутації становить 5%. Якщо розв'язок задачі не знаходився протягом 20 000 популяцій то вважалось, що розв'язок не знайдено.

Таблиця 1

Результати атаки на криптосистему

Розмір ключа (біт)	ГА з випадковим вибором батьківської пари, середня кількість обчислення фітнес-функції (відсоток успішних атак)	ГА з вибором батьківської пари методом рулетки, середня кількість обчислення фітнес-функції (відсоток успішних атак)	Груба сила, середня кількість обчислень фітнес-функції
10	1657 (93%)	944 (98%)	542
20	27813(37%)	17232(62%)	939925
50	-	61362(18%)	252280647282046

Для досягнення кращого результату розмір популяції був збільшений до 100 осіб, а коефіцієнт мутації залишився рівним 5% для ключів довжиною 10 і 20 біт, а для ключа довжиною 50 біт коефіцієнт мутації був зменшений до 3%, отримані результати відображені в таблиці 2.

Таблиця 2

Результати атаки на криптосистему

Розмір ключа (біт)	ГА з випадковим вибором батьківської пари, середня кількість обчислення фітнес-функції (відсоток успішних атак)	ГА з вибором батьківської пари методом рулетки, середня кількість обчислення фітнес-функції (відсоток успішних атак)	Груба сила, середня кількість обчислень фітнес-функції
10	586(97%)	428 (99%)	542
20	13102(92%)	5537 (97%)	939925
50	70368(14%)	37269 (24%)	252280647282046

Висновки. При проведенні експерименту використовувались два методи відбору хромосом батьків: метод рулетки та випадковий вибір батьків з батьківської популяції. Схрещення відбувалось за допомогою рівномірного кросинговеру, а для формування нової популяції використовувався елітарний відбір. Для того, щоб розв'язок задачі не зводився до локального мінімуму, була використана мутація. З вище наведених результатів видно, що збільшення розміру популяції та коригування коефіцієнта мутації призводить до швидшого вирішення задачі криптоаналізу. Отже, розкриття криптосистеми Меркле-Хеллмана за допомогою генетичного алгоритму є можливим і зазвичай займає менше часу, ніж розкриття повним перебором всіх можливих рішень.

Список літератури:

1. Merkle Ralph C., Hellman Martin E. Hiding information and signatures in trapdoor knapsacks // IEEE Transactions on Information Theory. – 1978. – Vol. 24, no. 5. – P. 525-530.

2. Гриник Р.О., Застосування генетичного алгоритму для вирішення задач крипто аналізу. / Гриник Р.О. // Інформаційно-комунікаційні технології в сучасній освіті: досвід, проблеми, перспективи : IV Міжнар. наук.-практ. конф., 21-22 жовт. 2015 р. : зб. наук. пр. – Ч. 1. – Львів, Вид-во ЛДУ БЖД, 2015. – С. 168-170

3. Назаренко А.А. Генетический криптоанализ шифра Меркле-Хеллмана / А.А. Назаренко, Р.Н. Кононенко // Известия ТРТУ, с.109-115

4. Панченко Т.В. Генетичні алгоритми// Издательский дом «Астраханский университет» 2007.

5. Елисеев Г. О. Применение биоинспирированных методов оптимизации для реализации криптоанализа классических и блочных криптосистем / Елисеев Г. О., Чернышев Ю. О. // Весник Пермского университета – 2010.

6. 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. 121–128

References:

1. MANPACK cryptosystem Merkle – Hellman. – [Electronic resource] – Access: https://en.wikipedia.org/wiki/Merkle%E2%80%93Hellman_knapsack_cryptosystem.

2. Grynyk R.O. Application of genetic algorithm for solving crypto analysis. / Grynyk R.O. // ICT in modern education: experience, problems and prospects: IV Intern. nauk. and practical. Conf., Oct. 21-22. 2015: Coll. Science. pr. – Part 1. – Lviv, Izdatel'stvo LSU BC, 2015. – P. 168-170

3. AA Nazarenko, RN Kononenko, Henetycheskyy cipher cryptanalysis Merkle-Hellman // Proceedings TRTU, s.109-115

4. Panchenko T.V. Genetic algorithms / Panchenko T.V. // Publishing Home "Astrahanskyy University" 2007.

5. Eliseev A.A. Application byoynspyryrovannыh optimization methods for cryptanalysis klassycheskyh Implementation and cryptographic / Eliseev A.A. Chernyshev YO // Vesnyk Perm University – 2010.

6. Sergeev A.S. / A.S. Sergeev, E.O. Dubrov // Theoretical and prykladnыe question sovremennыh information technology: Materials Vserossyyskoy 11 scientific-technical conference. – Ulan-Ude: VSHTU Publishing House, 2012. – P. 121-128.

❖❖❖