Genetic Algorithm Applied On Different Methods Of Cryptography

Geeta
Pursuing M.Tech (CSE)
SRM University, NCR Campus
Modinagar, India

Payal Pureha
Pursuing M.Tech (CSE)
SRM University, NCR Campus
Modinagar, India

Mr. Mohanraj Ramasamy
Asst.Professor (CSE)
SRM University, NCR Campus
Modinagar, India

ABSTRACT — GAs is a branch of artificial intelligence's stochastic search technique that is widely used in the field of optimization. They usually have four main elements which are an encoding structure that will be replicated, operators to affect the individuals of a population, a fitness function that indicates how good an individual is and a selection mechanism. Genetic algorithm (GA) is one of the main paradigms within EAs. They operate on a population of individuals, each presenting a possible solution to a given problem. Each individual is assigned a fitness score based on the fitness function. A selection mechanism selects highly fit individuals to reproduce the offspring by "cross breeding" (crossover) and mutation techniques. They are mainly designed to solve optimization problems. However, when cooperating with other techniques it can also deal with problems with constraints. It is so robust that it can be applied to a wide range of problem areas.

Keywords— **Genetic Algorithm, Cryptography, encryption, decryption, ciphers.**

INTRODUCTION

The application of a genetic algorithm (GA) to the field of cryptanalysis is rather unique. The primary goals of this work are to produce a performance comparison between traditional cryptanalysis methods and genetic algorithm based methods, and to determine the validity of typical GA-based methods in the field of cryptanalysis. Genetic algorithm can be applied to the various fields of cryptography.

IMPLEMENTION STEPS OF GENETIC ALGORITHM:

STEP 1:

The first step in the implementation of GA is encoding i.e. the representation of a problem solution/ chromosome. Encoding is mainly dependent upon the problem to be solved. Some of the different encoding techniques:

Binary Encoding

Chromosome 1:101100101100101011100101
Chromosome2: 111111100000110000011111

Permutation encoding

Chromosome 1: 8 9 7 4 6 2 3 5 1
Chromosome 2 : 2 3 1 4 8 5 6 7 9

Value encoding

Chromosome 1: 5.3243 1.2324 2.3293 0.4556 2.4545
Chromosome 2: HDIERJFDJDLDFABDJEIFLFEGT
Chromosome 3: (right), (back), (left), (back), (forward)

Tree encoding

STEP 2:

After the selection of chromosomes for crossover, the next task is to decide how to carryout the process of crossover so that genes from two parents can be recombined and children are created. The most common way of doing this is by random

selection and then exchange of genes, as shown in example, below.
Chromosome A=10011 | 10101110111
Chromosome B=**11001 | 01010011010**
Child A=10011 | **01010011010**
Child B=**11001** | 10101110111

STEP3:
Once the issue of crossover is resolved, the next step is mutation. "The main aim of carrying out mutation is to induce a certain level of diversity into population so that GA can be prevented from getting trapped into a local optimum"

(Bits selected for mutation are shown in bold)
Chromosome A=110**0**111000010010
Chromosome B=1101101**1**11110110
Mutated chromosome A=110**1**111000010010
Mutated chromosome B=110110**0**111110110

GA is a direct inspiration of the process of natural reproduction which consists of the processes of crossover, mutation and selection of chromosomes from one generation to another.

GENETIC ALGORITHM IN ENCRYPTION:

Cryptography is the science of making communication unintelligible to everyone except the intended receiver(s). It is the study of methods of sending messages in disguised form so that only intended recipients can remove the disguise and read the message. Cryptography offers efficient solution to protect sensitive information in a large number of applications including personal data security, internet security, diplomatic and military communications security, etc. through the processes of encryption/decryption. A cryptosystem is a set of algorithm, indexed by some keys(s), for encoding messages into cipher text and decoding them back into plaintext. The model for a secret key system, first proposed by Shannon is shown in Fig.

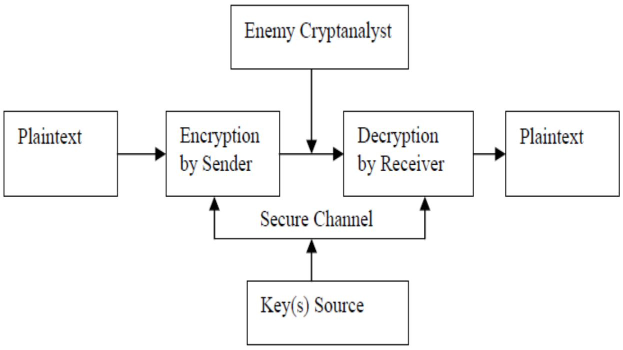

Fig. **Shannons model of secret communication**

A simple GA is mainly composed of three operations: selection, genetic, and replacement operation.

The overall procedure is summarized as follows:

Algorithm for Encryption
Step 1: Consider an image I (W*H) Where W and H are width and height of L Split the image I to a set of N vectors of length L where L=8 bytes.
Step 2: (crossover operation) for 1 = 0... N-l, each vector Vi from the set of N vectors: Do crossover we use secret key for crossover. In our research secret key have two attributes termed a, b belonging to 1 to 8. We do crossover by swapping a to b in each vector V1 [b0, b1, b2, b3, b4, b5, b6, b7] For example let the secret keys are 3 and 5 and we do crossover on vector V1 then V1 becomes [b1, b2, **b5**, b4, **b3**, b6, b7, b8]
Step 3: (mutation operation) For each vector Vi Do mutation by an another secret key of single variable of k. By the Vi [bk] = 255- V i[bk] For example let the secret key is 4 and we do mutation on vector V1 then V1 becomes [b1, b2, b5, (**255-b4**), b3, b6, b7, b8]
Step 4: Construct an encrypted image from the set of N vector that are produced from the mutation.

Algorithm for Decryption
Step 1: In the encrypted image Split the image in N vectors of L=8

Step 2: For each vector Vi Do the reverse mutation by secret key k i.e. Vi[bk]= 255 – Vi[bk] For example let the secret key is 4 and we do reverse mutation on vector V1 then V1 becomes [b1, b2, b5, **(255-(255-b4))**, b3, b6, b7, b8] Now V1 becomes [b1, b2, b5, **(b4)**, b3, b6, b7, b8]

Step 3: For 1 = 0... N-l, each vector Vi from the set of N vectors: Do reverse crossover by using secret key b to a We do crossover at this time by swapping b to a in each vector Vi Vi [b0, b1, b2, b3, b4, b5, b6, b7] For example let the secret keys are 3 and 5 and we do crossover on vector V1 then V1 becomes [b1, b2, **b3,** b4, **b5**, b6, b7, b8]

Step 4: Construct an image from the set of N vectors that are produced from the step 3.

GENETIC ALGORITHM APPLIED TO DIFFERENT CIPHERS:

In general, the genetic algorithm approach has only been used to attack fairly simple ciphers. Most of the ciphers attacked are considered to be classical, i.e. those created before 1950 A.D. which have become well known over time. Ciphers attacked include:

- Monoalphabetic Substitution cipher
- Polyalphabetic Substitution cipher
- Permutation cipher
- Transposition cipher
- Merkle-Hellman Knapsack cipher
- Vernam cipher

Monoalphabetic Substitution cipher

The simplest cipher of those attacked is the monoalphabetic substitution cipher. This cipher is described as follows:

• Let the plaintext and the ciphertext character sets be the same, say the alphabet Z

Let the keys, K, consist of all possible permutations of the symbols in Z

• For each permutation $\pi \in K$,

1. Define the encryption function $e_\pi(x) = \pi(x)$
2. And define the decryption function $d_\pi(y) = \pi^{-1}(y)$, π where π^{-1} is the inverse permutation to π.

Polyalphabetic Substitution cipher

This cipher is a more complex version of the substitution cipher. Instead of mapping the plaintext alphabet onto ciphertext characters, different substitution mappings are used on different portions of the plaintext. The result is called polyalphabetic substitution. The simplest case consists of using different alphabets sequentially and repeatedly, so that the position of each plaintext character determines which mapping is applied to it. Under different alphabets, the same plaintext character is encrypted to different ciphertext characters, making frequency analysis harder.

This cipher has the following properties (assuming m mappings):

• The key space K consists of all ordered sets of m permutations

• These permutations are represented as $(k_1, k_2, \ldots, k_m)$

• Each permutation k_i is defined on the alphabet in use

• Encryption of the message $x = (x_1, x_2, \ldots, x_m)$ is given by

• $e_k(x) = k_1(x_1)k_2(x_2) \ldots k_m(x_m)$, assuming the key $k = (k_1, k_2, \ldots, k_m)$

• The decryption key associated with $e_k(x)$ is $d_k(y) = (k^{-1}_1, k^{-1}_2, \ldots, k^{-1}_m)$.

Permutation/Transposition ciphers

Although simple transposition ciphers change the dependencies between consecutive characters, they are fairly easily recognized since the frequency distribution of the characters is preserved. This is true for ciphers considered, the permutation cipher and the columnar transposition cipher. Both ciphers apply the same principles, but differ in how the transformation is applied. The permutation cipher is applied to blocks of characters while the columnar transposition cipher is applied to the entire text at once.

Permutation cipher

The idea behind a permutation cipher is to keep the plaintext characters unchanged, but alter their positions by rearrangement using a permutation.

This cipher is defined as:

• Let m be a positive integer, and K consist of all permutations of $\{1, . . . ,m\}$

• For a key (permutation) $\pi \in K$, define:

• The encryption function $e_\pi(x_1, . . . , x_m) = (x_{\pi(1)}, . . . , x_{\pi(m)})$

• The decryption function $d_\pi(y_1, . . . , y_m) = (y_{\pi^{-1}(1)}, . . . , y_{\pi^{-1}(m)})$

Transposition cipher

Another type of cipher in this category is the columnar transposition cipher. As described in, this cipher consists of writing the plaintext into a rectangle with a prearranged number of columns and transcribing it vertically by columns to yield the ciphertext. The key is a prearranged sequence of numbers which determines both the width of the inscription

Knapsack ciphers

Knapsack public-key encryption systems are based on the subset sum problem, an Npcomplete problem. Given in this problem are a finite set S⊂ N+ (N+ is the set of natural numbers {1, 2, . . . }), and a target t 2 N+. The question is whether there exists a subset S'⊆ S whose elements sum to t[9]. For example:

- If S = {1, 2, 7, 14, 49, 98, 343, 686, 2409, 2793, 16808, 17206, 117705, 117993}
- And t = 138457
- Then the subset S0 = {1, 2, 7, 98, 343, 686, 2409, 17206, 117705} is a solution.

Merkle-Hellman Knapsack cipher

The Merkle-Hellman knapsack cipher attempts to disguise an easily solved instance of the subset sum problem, called a superincreasing subset sub problem, by modular multiplication and a permutation. A superincreasing sequence is a sequence (b1, b2, . . . , bn) of positive integers with the property that bi > Pi−1 j=1 bj , for each i, 2 ≤ i ≤ n. The integer n is a common system parameter. b1, b2, . . . , bn is the superincreasing sequence and M is the modulus, selected such that M > b1 + b2 + . . . + bn. W is a random integer, such that 1 ≤ W ≤ M − 1 and gcd(W,M) = 1. π is a random permutation
of the integers {1, 2, . . . , n}. The public key of A is (a1, a2, . . . , an), where ai = Wb(π i) mod M, and the private key of A is (π,M,W, (b1, b2, . . . , bn)).
The steps in the transmission of a message m are as follows (B is the sender, A is the receiver):

1. Encryption - B should do the following:

• Obtain A's authentic public key (a1, a2, . . . , an)

• Represent the message m as a binary string of length n,

• m = m1m2 . . .mn

• Compute the integer c = m1a1 + m2a2 + . . . + mnan

• Send the ciphertext c to A

2. Decryption - To recover the plaintext m from c, A should do the following:

• Compute d = W−1c mod M

• By solving a superincreasing subset sum problem, find the integers r1, r2, . . . , rn,
ri 2 {0, 1} such that d = r1b1 + r2b2 + . . . + rnbn.

• The message bits are mi = r π (i), i = 1, 2, . . . , n.

Vernam cipher

The Vernam cipher is a stream cipher defined on the alphabet $A = \{0, 1\}$. A binary message $m_1m_2 . . . m_t$ is operated on by a binary key string $k_1k_2 . . . k_t$ of the same length to produce a ciphertext string $c_1c_2 . . . c_t$ where $c_i = m_i \leq k_i$, $1 \leq i \leq t$. If the key string is randomly chosen and never used again, this cipher is called a one-time system or one-time pad. The one-time pad can be shown to be theoretically unbreakable. If a cryptanalyst has a ciphertext string $c_1c_2 . . . c_t$ encrypted using a random, non-reused key string, the cryptanalyst can do no better than guess at the plaintext being any binary string of length t. It has been proven that an unbreakable Vernam system requires a random key of the same length as the message.

SUMMARY

The primary goal of this research is to apply the genetic algorithm in different methods of cryptography. There are thee steps involving in genetic algorithm to be applied on any method as encoding, crossover and mutation. Genetic algorithm is applied to get the best result among different solution for a problem.

Here it being applied to encryption where an image is used to encrypt the information. The image has different segments to encrypt it and further used to decrypt it with the same image. It can also be applied to different ciphers to produce a performance comparison between traditional cryptanalysis methods and genetic algorithm (GA) based methods, and to determine the validity of typical GA-based methods in the field of cryptanalysis. The main metric for both classical and genetic algorithm attacks was elapsed time. In the classical algorithm case, the elapsed time was comprised almost completely of time spent interacting with the user, while in the genetic algorithm case, it was comprised almost entirely of processing time. This difference in composition of elapsed time makes it difficult to compare the attacks on a time basis. Therefore, an additional metric was required. The secondary metric used was the percentage of successful attacks per test set. Success was defined as complete decryption of the ciphertext. This metric was only calculated for the GA-based attacks, as a classical attack will run until the text is decrypted. A GA-based attack, on the other hand, runs for a specific number of generations, independent of the decryption of the ciphertext. Using this metric, as well as elapsed time, gives one a better feel for the usefulness of each attack. The time measurement is irrelevant if the attack is unsuccessful. These two metrics were chosen so that traditional and GA-based attacks could be compared. Elapsed time measures the efficiency of the attack while the percentage of successful attacks measures how useful the attack is. Both of these metrics are needed to gain a complete picture of an attack. The two transposition attacks were tested using much larger population sizes and numbers of generations, while the permutation attack was attempted on larger block lengths. The two transposition cipher attacks obtained about the same success rate as before, while the permutation cipher attack experienced decreased success. Since the scoreboard approach was successful in both cases it was used, this style of GA-based attack was attempted on the monoalphabetic substitution cipher.

CONCLUSION AND FUTURE USE

Genetic algorithm provides the optimization among the different solutions as it provide the best among all the solutions. It can be expanded to further more methods of cryptography like elliptic curve cryptography. In the future work, there is a planning to design sophisticated software based on this technique of genetic algorithms and its different application areas

REFERENCES

[1] David E Goldberg, „Genetic algorithms in search, optimization and machine learning" , Addison- Wesley Pub.Co.1989.

[2] Alcott, Louisa May. *Little Women Parts I & II*. Great Literature
Online 1997-2003. Retrieved August 20, 2003 from http://www.underthesun.cc/Classics/Alcott/littlewomen/.

[3] Douglas Stinson, *Cryptography: Theory and Practice*, *Third Edition*, CRC/C&H, 2006.

[4] William Stalling, "Cryptography and Network Security: Principles and Practice", 2/e, prentice hall, 1999.