Swarm Intelligence and Evolutionary Computation based Cryptography and Cryptanalysis of 4-round DES algorithm

Anjali Dadhich[#1], Dr. Surendra Kumar Yadav*[2]
[#](M.tech Student) Computer Science Department, [*2](Associate professor) Computer Science Department,
JECRC University Jaipur

Abstract— Over the past decade, there has been an increasing research in the application of Fuzzy Logic and Evolutionary Computation methods to problems in the field of cryptography and cryptanalysis. This is primarily due to the effectiveness of application of these methods to handle hard problems, and to the resulting automated designs pertaining to cryptanalysis of cryptosystems. This paper begins with a brief introduction to cryptography and fuzzy-evolutionary computation methods. A short survey of the applications of these computational intelligence techniques to cryptographic problems follows, and then our contribution is presented. Specifically, we have viewed some cryptographic problems as discrete/continuous optimization problems and are addressed using Evolutionary Computation methods, particularly swarm intelligence and particle swarm optimization. Furthermore, the effectiveness of Swarm Intelligence to optimize the search space of some of the cryptographic functions is studied. Finally, theoretical issues of image cryptography are presented. The experimental results suggest that the discrete optimization problem formulation and representation are critical factors that determine the performance of Evolutionary Computation methods to cryptography. Moreover, strong cryptosystems must not reveal the inherent patterns of the encrypted messages, their decryption keys and the encryption algorithm structure, it is reported that swarm intelligence and evolutionary computation methods constitute a strong measure of the cryptosystems' security.

***Keywords*— Cryptography, cryptanalysis, cryptosystem, evolutionary computation, computational intelligence, encryption, decryption, swarm intelligence.**

I. INTRODUCTION

Cryptography is defined as the transformation or encryption of a given information source or message into another message that reads differently, and appears meaningful only to the intended recipient upon decryption. The message which is subjected to encryption is known as the plaintext (or cleartext), and the transformed message is known as the ciphertext. Cryptanalysis is the process of recovering the plaintext from the ciphertext without having the knowledge of the decryption key. A cipher is defined as a cryptographic algorithm which is a mathematical function that enables the encryption and decryption of messages [21]. Ciphers are usually classified into two categories: the symmetric-key ciphers and the public-key ciphers. In symmetric-key, the sender and the receiver secretly agree upon a key that is used for encryption and decryption. This type of cryptosystems suffer from the drawback that before sending the message, these require the key to be communicated between the sender and the receiver through a secure channel. If eavesdropped, a third person could intercept the key and thus decrypt the message illegally, or could modify the key so that the message could not be retrieved by the receiver [3]. Public-key ciphers alleviate this drawback as they are designed in a manner such that the key used for encryption is publicly available, but it differs from the key used for decryption. The key used in decryption is confidential, and is kept secret. The effectiveness of public-key cryptography lies in the fact that although the two keys are functionally interrelated with one key available to everyone; yet the computation of the secret from the known public key is computationally unfeasible as well as intractable [5].

Thus, an encrypted message can be sent by anyone using the public key, but the message can be decrypted only by the owner of the secret key.

Cryptology comprises of two main problems

(a) Cryptography: This branch of study looks for designing unbreakable cryptosystems.

(b) Cryptanalysis: This branch of study looks for designing methods to break secure cryptosystems [21].

The purpose of this paper is to give a survey of cryptographic applications that can be developed with the help of evolutionary computation methods, and to address their applicability to the real-world scenarios. This paper aims to investigate the applicability of Computational Intelligence (CI) techniques such as particle swarm optimization and Evolutionary Computation to solve cryptology problems, their advantages and improvements over other methods and the future perspectives. Evolutionary Computation methods are inspired from natural and biological evolution theory and provide following characteristics [7]:

(a) Copes well with large and randomly defined search spaces.

(b) Computational speed is high and cost is comparable to other techniques

(c) Some EC examples are Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Differential Evolution (DE).

This paper structured as follows: In section 2 we discussed overview of cryptographic algorithms. Section 3 discusses an

ISSN: 2278 – 1323

overview of evolutionary computation techniques and highlights particle swarm optimization that we are using in this work. In section 4 fuzzy set theory is discussed and in section 5, we discussed the related work. The proposed algorithm and results is given in subsequent sections and at last conclusion and future scope is given.

II. OVERVIEW OF CRYPTOGRAPHIC ALGRITHMS

The popular cryptographic algorithms that are used in cryptanalysis experiments are listed as follows:

Block Cipher: - A block cipher maps n-bit plaintext blocks to n-bit ciphertext blocks, where n is a decided block-length, where the cipher function is parameterized by a k-bit key K. A subset K, which is the key space of the set of all k-bit vectors, is set as the look up table for selecting a key. The block cipher function must allow unique decryption through either symmetric-key or public-key [20]. A widely used type of block cipher is a Feistel cipher, which is an iterated block cipher based on repetitive computation of simple functions. The simple functions are known as *round functions*, on the input data that are used for a predetermined number of rounds. The resulting function maps an n–bit plain text to a ciphertext. In a Feistel cipher, the currently computed n–bit word is divided into (n/2) bit blocks, which are labelled as the left part L_i and the right part R_i [8].

$$L_i = R_{i-1}, \quad R_i = L_{i-1} \oplus F_i(R_{i-1}, K_i), \quad \ldots\ldots (1)$$

In the ith round, $1 \leq i \leq r$, the subkey K_i, derived from the cipher key K, is used along with F_i which is an arbitrary round function. After the last round of the function has been executed, the two halves are swapped and the outcome is the ciphertext C of the Feistel cipher, i.e. $C = (R_r, L_r)$. The encryption procedure of Feistel ciphers is illustrated in Fig. 1. The sub keys K_i and the round functions F_i are applied in reverse order which make the Feistel structure an attractive choice for software and hardware implementations [9].

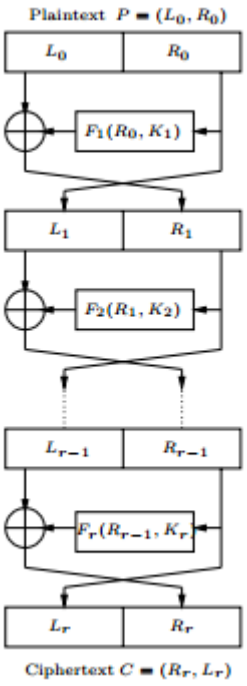

Figure 1 Feistel Block cipher tehnique [9]

Data Encryption Standard and S-boxes: - One of the most widely used Feistel ciphers is the Data Encryption Standard (DES) algorithm [22]. DES is a symmetric-key cryptosystem, where the parties exchanging information possess the same key. It encrypts plaintext blocks of n = 64 bits, resulting in 64 - bit ciphertext blocks. The effective key size is k = 64 bits, out of which 8 bits are used as parity bits. The main component of the round function in DES is the F function working on the right half of the data using a subkey of 48 bits and eight S-boxes. S-boxes are defined as nonlinear *mappings* that transform 6 bits into 4 bits and contain the nonlinear component of DES. The 32 - bit output of the F function are XORed with the left half of the data and the two halves are exchanged. A detailed description of the DES algorithm can be found in [22].

DES has been prone to cryptanalysis attacks over the past two decades and two of the most powerful cryptanalysis attacks on DES and Feistel based ciphers rely on the exploitation of specific weaknesses of the S-boxes. These attacks are known as the Linear Cryptanalysis and the Differential Cryptanalysis, which were successfully applied to the cryptanalysis of DES [20].

Differential Cryptanalysis: - Differential Cryptanalysis (DC) is a chosen plaintext attack where the opponent has a temporary access to the encryption function and can construct ciphertexts for some chosen plaintexts. DC studies and analyzes the effect that the specific differences in the plaintext pairs have on the differences in the resulting ciphertext pairs. These differences are then used to assign probabilities to possible keys which help to identify those bits of the key that were used in the encryption process. This method works on the pairs of plaintexts having a specific difference and relies on the resulting ciphertext pairs. In DES, the difference is

ISSN: 2278 – 1323

usually chosen to be a fixed XORed value of the two plaintexts [13].

To locate the most probable key any chosen pair of encrypted plaintexts is associated with the XOR value of its two plaintexts. The XOR value of the corresponding ciphertexts and the XOR values of the inputs of each encryption round form an r-round characteristic [1]. An r-round characteristic in DES cryptanalysis is defined as a tuple (row-vector) $\Omega = (\Omega_P , \Omega_\Lambda, \Omega_C)$; where Ω_P and Ω_C are n - bit numbers and Ω_Λ is a list of r elements defied as $\Omega_\Lambda = (\Lambda_1, \Lambda_2,...,\Lambda_r)$. Each element in Ω_Λ is of the form $\Lambda_i = (\lambda_i I, \lambda_i O)$, where $\lambda_i I$ and $\lambda_i O$ are n/2 bit numbers and n is the block size of the cryptosystem []. Each combination of the plaintext XOR, Ω_P, Ω_C, Λ_i and the ciphertext XOR allows the search for a particular set of bits in the subkey of the last round. These bits characterise the ones that enter particular S-boxes based on the chosen characteristics. The characteristics are chosen based on a maximal probability and a maximal number of subkey bits whose occurrences can be counted. DC is a very powerful statistical cryptanalysis method that rarely fails. A more extended analysis on DC and its results on DES for various rounds is provided in [6]. DC was the first theoretical cryptanalysis for DES that required lesser number of steps than the brute force attack. DC, on an average, tested all 2^{56} possible keys [3].

Public key Cryptography schemes: - The Public key cryptography is closely related to various complex mathematical problems in the field of computational algebra, mathematical logic, number theory, algebraic geometry, probability theory, Diophantine complexity and many more. The common phenomena in all these problems are the *factorization* and the *discrete logarithm*. These cryptosystems are based on the assumption that the corresponding cryptanalysis problems are computationally intractable in the sense that their computation cannot be completed in polynomial time [1].

Discrete Logarithm Problem (DLP): - DLP is a cryptanalysis algorithm which results in an efficient algorithm for the computation of an integer x that satisfies the relation $\alpha^x = \beta$, where α is a fixed primitive element of a finite field F_q. β is a non–zero element of the same field. It is assumed that x is the smallest non-negative integer with $\alpha x = \beta$, where x is called the index or the discrete logarithm of β. In case of a finite field Z_p of prime order p, a primitive root g modulo p is selected where $g^u = h \pmod p$ [1].

Elliptic Curve based Cryptosystems: - Cryptographic systems based on elliptic curves were proposed as an alternative to public key cryptosystems. The main advantage offered by these cryptosystems is the use of smaller parameters compared to the conventional cryptosystems (e.g. RSA, AES or DES). This is primarily due to the increased difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP), which requires more time to solve than the time required for the solution of corresponding finite field in the DLP. The security of cryptosystems that rely on discrete logarithms is based on the assumption that these problems are not solvable in polynomial time [1] [3].

III. OVERVIEW OF EVOLUTIONARY COMPUTATION

Evolutionary Computation (EC) algorithms present a range of problem-solving techniques based on principles of biological evolution such as natural selection and genetic inheritance. These algorithms have been applied to solve a variety of difficult problems, and have also been extended to cryptography. Examples of such techniques include the evolving hash functions or creation of a new block cipher. Initial applications of EC to cryptography have emerged over two decades ago, and in recent years there has been an increased interest in this area. Still, some areas such as problem formulation and representation remain open to research [6] [7].

Alan Turing first conceived the idea of Artificial and Computational Intelligence in as early as 1950, when he hypothesized that computers can mimic the processes of the human brain. This hypothesis relied on the assumption that any reasoning can be carried out on a large enough deterministic computer. Turing's hypothesis has inspired a vast amount of research in the effort to embed intelligence into computers. Computational Intelligence (CI) can be considered as the study of adaptive mechanisms that enable intelligent behavior of a system in complex and changing environments [6]. These mechanisms display the ability to learn the environment, become aware of the context and adapt to new situations. To enable intelligent behavior, CI systems model some aspects of biological and natural intelligence. Thus, CI systems are usually hybrids of paradigms such as Evolutionary Computation systems, Artificial Neural Networks and Fuzzy systems, supplemented with elements of reasoning. The basic elements of Evolutionary Computation are describes as follows [13]:

Evolutionary Computation: -. Evolutionary Computation (EC) is a branch of CI that is based on biological and evolutionary mechanisms such as natural selection and adaptive behavior. EC finds extensive use in optimization and classification methods. Natural selection refers to the survival of the fittest where an offspring must retain those characteristics of its parents that are best suited to survive in a given environment. Weaker offspring lose the battle of survival. The EC paradigms that form this class are Genetic Algorithms (GA), Genetic Programming (GP), Evolutionary Programming (EP), Evolution Strategies (ES) and Differential Evolution (DE) [13]. The social and adaptive behavior of animals that behave in a synchronized and organized manner in groups inspired the development of another class of EC methods, namely Swarm Intelligence (SI). These methods model organisms that are organized into groups and act for a common cause. Typical examples are the search for food mechanisms of bird flocks, schools of fish and ant colonies. The study of many biological processes of social and adaptive behavior led to the opinion that social sharing of information among the individuals of a population can generate an evolutionary advantage [17]. Paradigms that belong to this class are the Particle Swarm Optimization (PSO) method and the Ant Colony Optimization (ACO) method.

ISSN: 2278 – 1323

Genetic Algorithms: - The biological experimentation in simulating natural genetic systems using computers gave rise to Genetic Algorithms (GA), invented by John Holland. He combined machine intelligence and machine learning with the abilities of GAs to artificial computer systems [2]. These systems display an ability to adapt to environmental changes and also exhibit self–adaptation such that they could adjust their operations according to their interaction with the environment.
The innovation in GA is that the problem usually consists of a population of individuals for the search procedure instead of a single search point, based on natural evolution and genetic inheritance. As in natural evolution, each biological species has to search for the most appropriate adaptations to a complex and changing environment to ensure its survival. The individuals of the population are called chromosomes or genotypes. Each chromosome consists of parts called genes and is responsible for the inheritance of one or more characteristics [4]. The evolution procedure of a population of chromosomes corresponds to a search on the space of possible problem solutions and has to balance between two different scopes, the exploitation of the best solutions and the exploration of the search space. The evolution procedure of GAs is implemented using crossover and mutation operators. These operators alter chromosomes to produce better ones. The selection of the new population is done using a fitness function. GAs usually represent chromosomes using binary representation, but GA methods that use other arithmetic systems, including floating point numbers, have also been developed. GAs have been successfully applied to optimization problems arising in different fields such as applied mechanics and design, time-scheduling, the traveling salesman's problem, optimal control and robotics, and economics among others [23].

Evolutionary Programming: - Evolutionary Programming (EP) was developed by Larry Fogel with the aim to predict the changes of the environment. The environment in EP is described as a sequence of symbols from a finite set and the evolution algorithm provides as output a new symbol. This symbol maximizes the fitness function that describes the accuracy of the prediction. Evolutionary Programming uses the principle of the selection of the fittest for the new population, but only the mutation operator is used for altering the individuals of the population. To this initial version of EP two more basic concepts have been added. These concepts pertain to the ability of handling continuous parameters in addition to the discrete ones, and the ability of self-adaptation. EP can address optimization and classification problems with applications in several fields [24].

Evolution Strategies: -. Ingo Rechenberg and Hans-Paul Schwefel used the idea of mutation to obtain the optimal design for a sequence of joints in a liquid transition pipe. The classical optimization techniques based on the gradient of the fitness function were unable to handle the problem and therefore they tried experimentation with mutation. Using mutation they caused a small perturbation to the best existing solutions to explore the neighborhoods in the search space of the problem, in a stochastic manner. This experimentation laid the foundation for the beginning of the development of Evolution Strategies. Evolution Strategies can be considered as evolutionary programs that use floating point representation and employ a recombination and a mutation operator. They have been used for the solution of several optimization problems with continuously changing parameters and they have been recently extended for discrete problems.

Genetic Programming: -. Genetic Programming (GP) was developed by Koza [25]. The idea behind GP was that instead of constructing an evolutionary program to solve the problem, one could locate the most proper solution for the specific problem, in the space of computational programs. A population of executable computational programs is created and every individual program competes with the rest. The non efficient programs become idle while the best ones reproduce by means of operators such as crossover and mutation. The evaluation of the programs is done using fitness measure on a pre-defined set of problems [6] [7].

Tabu Search: - Tabu search was invented by W. Glover. Tabu search is an optimization technique in the class of local search techniques that enhances the performance of a local search method. Tabu search uses memory structures where it stores the solutions. The most important of memory structures is a tabu list. Once the potential solution is determined, it is put on a tabu list so the algorithm cannot visit that possibility repeatedly. A tabu search uses a local search procedure to iteratively move from one solution to another which is in the proximity of the first solution. To explore regions of the search space that would be left unexplored, tabu search modifies the local structure of each solution as the search progresses [6] [7].

Simulated Annealing: - Simulated annealing, invented by S. Kirkpatrick, is a generic probabilistic meta-heuristic inspired by the cooling processes of molten metal. Simulated annealing combines hill-climbing technique with the probabilistic acceptance of non improving moves. In simulated annealing, each particle in the search space is analogous to a state of a physical system, and the function that needs to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system from arbitrary state to the state with minimal energy. The search starts at some initial state with a control parameter known as the temperature. The search tries to avoid local minima by jumping out of them early in the computation. Toward the end of the computation, when the temperature, or probability of accepting a worse solution, is nearly zero, the search simply seeks the bottom of the local minimum. The chance of getting a good solution can be traded off with computation time by slowing down the cooling schedule. The slower the cooling, the higher is the

ISSN: 2278 – 1323

chance of finding the optimum solution, but the run time is also longer [4].

Particle Swarm Optimization: - Particle Swarm Optimization (PSO) method is a population-based algorithm that exploits a population of individuals, to identify promising regions of the search space. The population is called swarm and the individuals are called particles. Each particle moves with an adaptable velocity within the search space, and stores the best position it encountered. In the global variant of the PSO the best position ever attained by all individuals of the swarm is communicated to all the particles. In the local variant, each particle is assigned to a neighborhood consisting of a pre-specified number of particles. In this case, the best position ever attained by the particles that comprise the neighborhood is communicated among them. Optimization is the act of obtaining the best result under given circumstances [6]. PSO is an evolutionary optimization technique introduced by Kennedy and Eberhart in 1995. Since then it has been widely used to solve a wide range of optimization problems. The PSO concept is based on social behaviour of bird flocking and fish schooling. In the PSO algorithm, the particles move around in the multi-dimensional search space. The positions of individual particles are adjusted according to their previous best positions and the neighbourhood best or the global best. PSO is the only evolutionary algorithm that does not implement the survival of the fittest. The reason is that all particles in PSO are kept as members of the population during the course the searching process. Therefore, as mentioned in [7] it can conclude that the PSO algorithm can be utilized to a wide range of continuous optimization problems. PSO is an artificial intelligence technique that can be used to find approximate solutions to extremely difficult numeric maximization and minimization problems [7]. PSO is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions. It considered as a mid-level form of artificial life (A-life) or biologically derived algorithm, occupying the space in nature between evolutionary search, which requires eons, and neural processing, which occurs on the order of milliseconds. It uses the concept of fitness, as do all evolutionary computation paradigms. Unique to the concept of PSO is flying potential solutions through hyperspace, accelerating towards the better/optimum solution. Other evolutionary computation schemes operate directly on potential solutions which are represented as locations in hyperspace [6]. PSO has been enormously successful. The first practical application of PSO is in the field of neural network training. Many more areas of application have been explored, including telecommunications, control, data mining, design, combinatorial optimization, power systems, signal processing, and many others.

Ant Colony Optimization: - The Ant Colony Optimization (ACO) algorithm is a Swarm Intelligence method for tackling, in general, Combinatorial Optimization problems, like the traveling salesman problem and telecommunications scheduling. It exploits a population of members called artificial ants and it has been inspired from experiments with real ant colonies. In these experiments it was discovered that after a small time interval groups of ants choose the shortest between two routes to transfer food to their nest. This ability becomes possible by a chemical substance, called pheromone, which ants leave in the environment that serves as an indirect communication mechanism. Thus, at the beginning the route chosen by ants appears to be random but with the progress of time the possibility of choosing the shortest path becomes higher as the quantity of pheromone on this path increases faster compared to the quantity of pheromone on longer paths. This simple idea is implemented by the ACO methods to locate solutions and address hard optimization problems.
The foraging behavior of ants and their ability to find the shortest path to a food source has led to the creation of a highly popular algorithmic model known as Ant Colony Optimization (ACO). The foraging behavior of ants is shown in Fig 2-5. ACO is one of the most common Swarm Intelligence (SI) techniques applied to cryptography. ACO is modeled on the behavior displayed by the collective group of ants in locating the nearest and richest sources of food without any individual ant having knowledge about it [11]. Ants lay a chemical substance called pheromone to leave a trail of the routes while moving. The piled up traces of pheromone evaporate with time and so the longer paths hold thin or no traces of pheromone after some time, leading ants to choose the shorter path where stronger traces of pheromone are still present. The thickness of pheromone deposition on a path indicates the frequency and intensity its use. Stronger pheromone concentrations encourage more ants to take that path, reinforcing the pheromone concentration along that path. Ants who first locate a food source return to their nest earlier compared to others. The pheromone concentration on this path is stronger than on longer paths, in turn leading more ants to take the shorter path. Another magnificent facet demonstrated by the ants is the ability to sort objects. Individual ants wander in a random direction and identify different type of objects. Any object encountered that is not in the ant environment is carried. The similar objects are also carried and deposited together as a heap. Researchers have applied this fascinating ant metaphor to the intrusion detection domain. Artificial ants are ants with memory, have preference for trails with larger amounts of pheromone, and this pheromone trail acts as an indirect communication system between artificial ants to find the best path [1] [17].

Figure 2 Real ants follow a path between nest and food source[1]

ISSN: 2278 – 1323

Figure 3 An obstacle appears on the path: Ants choose whether to turn left or right with equal probability [1].

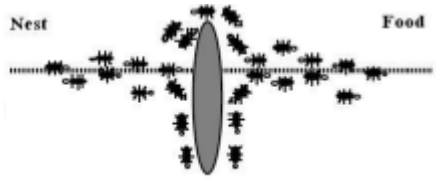

Figure 4 Pheromone deposited more quickly on the shorter path [1]

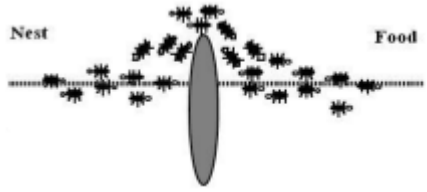

Figure 5 All ants have chosen the shorter path [1]

Different colonies of ants deposit different types of pheromone and that pheromone attracts ants from the same colony. This is a marvelous trait harnessed by computational intelligence researchers to design IDS based on ACO, using individual single ant colonies to locate all classes of attacks (paths) [5]. Multiple ant colonies leaving strong pheromone concentrations on a path is analogous to an alarm that signifies the presence of a malicious activity on the network. The architecture with more than one colony of ants dedicated to looking for solutions to multi-class classification problems, with each ant colony dedicated to detect one class at a time is also widely accepted.

IV. OVERVIEW OF FUZZY SET THEORY

Traditional set theory and binary-valued logic both require two values of parameters, be part of a set or not, and 0 or 1, respectively. Human reasoning, however, includes a measure of uncertainty, and hence is not exact. Fuzzy sets and fuzzy logic allow what is referred to as approximate reasoning. With fuzzy sets, an element belongs to a set with a certain degree of certainty. Fuzzy logic allows reasoning with these uncertain facts to infer new facts, with a degree of certainty associated with each fact. In a sense, fuzzy sets and fuzzy logic allow the modelling of common sense [22]. The uncertainty in fuzzy systems is referred to as non statistical uncertainty, which should not be confused with statistical uncertainty. Statistical uncertainty is based on the laws of probability, whereas non statistical uncertainty is based on vagueness, imprecision and/or ambiguity. Statistical uncertainty is resolved through observations. As quoted by Henri, E. (1954) that 'Precision is not truth', i.e. every day to day activity is associated with some kind of imprecision and Fuzzy Logic addresses that uncertainty inherent in the information and solution models are designed with a dedicated characterization of that uncertainty [15]. For example, when a coin is tossed we are certain what the outcome is, while before tossing the coin, we know that the probability of each outcome is 50%. Non-statistical uncertainty, or fuzziness, is an inherent property of a system and cannot be altered or resolved by observation. Fuzzy systems have been applied to control systems, gear transmission and braking systems in vehicles, controlling lifts, home appliances, and controlling traffic signals.

V. RELATED WORK: REVIEW OF CRYPTOGRAPHY AND CRYPTANALYSIS USING EVOLUTIONARY COMPUTATION TECHNIQUES

ICIGA (Improved Cryptography Inspired by Genetic Algorithms) system represents an improvement of a system "*Genetic Algorithms Inspired Cryptography*" [16]. ICIGA is a block cipher system where secret key is generated in each session via a random process. The length of the key and the block size are the parameters adjustable by the user. Based on the key length, plaintext is divided into the parts of the same size. The first part is used to generate secret key which will be used in ciphering of the message. The difference between the two versions of the system is in adding more transformation operations in the ICIGA system. The algorithm for encryption can be described as follows:

(a) First, break the binary encoded plaintext into parts of equal size based on the length of the key, and then break those parts in the blocks of the same size.

(b) Next, crossover and mutation operations based on the genetic algorithms are applied. Those operations work on randomly selected blocks and positions in those blocks. Based on the trace of those operations a secret key is generated. Then mask the position of the genetic algorithm operators by applying left shift.

(c) Third, apply left shift on whole part to mask the distribution of the blocks.

(d) Finally, apply same steps to the rest of the plaintext but use secret key to choose genetic algorithm operations and positions instead. Decryption is composed of the right shift operation and the same genetic algorithm operators since the inverse of left shift is right shift, and the crossover and mutation are involutions. The operations in deciphering process should be done in reverse order. The crossover operation is done by permuting the bits between two blocks, and the mutation is done by applying logical negation on the bits. Crossover operation can be regarded CBC (Cipher Block Chaining) mode of ciphering and mutation operation as ECB (Electronic Code Book) mode of ciphering. It is necessary to mention that no actual genetic algorithm is used, and because of that, no selection scheme or fitness function is needed. Crossover and mutation operations that draw inspiration from genetic algorithms are used for encryption and decryption. When compared the ICIGA system with other

ISSN: 2278 – 1323

symmetric key ciphers - DES, IDEA, and AES algorithms, ICIGA resulted in faster conversion times and better/stronger key retaining [15] [16].

Many security protocols such as hash functions are usually designed by experienced experts from the area of cryptography, but it is also possible to develop them from automatically obtained nonlinear functions. To develop a block cipher, it is necessary to decide on highly-nonlinear functions that will be used in that algorithm. As criteria for estimating nonlinearity of a function avalanche effect is used. Informally, avalanche effect can be defined as the effect that minimum change of the input (one bit) changes on average, half of the output bits. The core of their work was to design functions with nearly ideal amount of avalanche effect. As an evolutionary computation method in developing functions with that kind of properties, authors used genetic programming. Experiments were conducted after an adequate parameter set was chosen. For a function set, efficient operations that are easy to implement in software and hardware were chosen. To develop a block cipher that. S-boxes or substitution boxes are generally n-input, m output functions. Additionally, that can be viewed as a combination of m individual single output Boolean functions. In some applications, substitution boxes are formed by simple Boolean functions which take several Boolean inputs and give a single output [12].

There exist many ways to construct S-boxes. Generally, they can be divided the random way and wit the mathematical methods. Mathematical methods provide good cryptographic properties, but it can be vulnerable to algebraic attack if the expression is too simple. The random way can be to randomly generate S-boxes and test whether they are good. Another possibility is to construct S-boxes on the basis of some previously known S-boxes that have good properties. A cost function can be developed for single output Boolean function and generalize it for the use in S-boxes. Cost function represents the condition that needs to be satisfied, and that is the approach that authors adopted in [3]. Formal criteria for the single output Boolean functions are to have high nonlinearity and low autocorrelation. Those criteria are selected because they provide some level of protection against linear cryptanalysis, and differential cryptanalysis. In year 2000, a new cost function family was proposed that offer significant improvement for the single output Boolean function case. That cost function defined the cost over the whole Walsh-Hadamard spectrum rather than on extreme values as it was done prior to that. The single output cost functions can be applied to each function defined as a linear combination of the outputs. The search starts with regular, but randomly chosen function and moves around the search space. Simulated annealing was chosen to found the best solution. Annealing based search is used to minimize the value of the new cost function, and then hill-climb from the best solution with respect to nonlinearity to produce final solution. At the end, it is necessary to measure the nonlinearity, autocorrelation, and algebraic degree of the final solution. The results obtained by this method are better than the results obtained in the case of human made S-boxes for the case when functions have small number of inputs. Furthermore, for some cases when number of inputs was larger, the results were also better than those obtained by human made S-boxes [8].

VI. APPLYING EVOLUTIONARY COMPUTATION IN CRYPTANALYSIS

We propose to apply EC, particularly SI to cryptographic problems derived from classical public key cryptosystems which are first formulated as discrete optimization tasks and EC methods are applied to address them. We define cryptanalysis as a Discrete Optimization Task and the EC algorithm, namely PSO method is applied for the cryptanalysis. The reported results suggest that the formulation of the problems as discrete optimization tasks preserves their complexity which makes it difficult for the methods to extract pieces of information [9].Cryptanalysis is one of the major challenging areas of intense research in the discipline of computer and data security. It is a process of looking for the weakness in the design of the cryptosystem or hash algorithms. Evolutionary computation (EC) is a branch of the computational intelligence; EC algorithms have got significant importance in determining efficient solutions of different complex and active problems like cryptanalysis. EC algorithms are stochastic optimization methods that involve algorithmic mechanisms inspired by natural evolution and social behaviour. These methods have been proved to be efficient and effective where deterministic optimization methods fail and can handle problems that are involve discontinuous objective functions and disjoint search spaces. This thesis invades different types of encryption and hash function algorithms on different types of data. First, the cryptanalysis of Data Encryption Standard (DES) was studied where two approaches are proposed based on particle Swarm Optimization Algorithm (PSO). These algorithms are known cipher-text attack for four round DES and known plain-text attack of DES-16 algorithm [23].

VII. PROPOSED ALGORITHM, RESULTS & DISCUSSION

This section presents our results obtained from the application of EC and Swarm Intelligence methods in the cryptanalysis of known cryptosystems. The algorithm we have used is described as follows:

ISSN: 2278 – 1323

Algorithm Particle swarm optimization cryptanalysis algorithm

Initialize algorithm variables: G the maximum number of generations to consider, N the solution pool size and any other problem dependent variables
Initialize the value of the weight factor w
Generate initial population of N particles
Input the cipher text

1: **For** G iterations do
2: Known cipher-text is decrypted using the keys
3: Calculate the fitness value of each key according to equation (3.2.1)
4: Select particle neighbors to update its velocity
5: Update the position of *gbest* and *pbest*
6: **if** fitness (X_{id}) < Fitness $pbest_{id}$ then $pbest_i = X_{id}$
7: **else if** fitness (X_{id}) < Fitness $gbest_d$ then $gbest_i = X_{id}$
8: **End if**
9: Update velocity and position of each particle according to equations (2.3.1) and (2.3.2)
10: Update the value of the weight factor
11: Repeat until the optimum key is found or maximum number of generations have been reached
12: **End for**
13: Repeat the steps from 1 to 12 for T time trials

As mentioned in the previous section, PSO is a population-based search algorithm that applied over a population of individuals and returns a region of the function space with best possible solution. The population is known as the swarm and the individual entities are termed as particles [6]. Each particle is moved in the search space at an adaptive velocity. Each particle is clustered into a neighbourhood that consists of a specific number of particles. Each particle retains the best position it went through in its memory. Finally, the best position encountered by all the particles is communicated to each particle of the swarm [6].

The particles represented a multidimensional space encoded as a string of positions with all the dimensions independent of each other [16]. Though the dimensions are independent of each other, the particles in our cryptanalysis problem are not independent of each other. Therefore a solution in the search space represented as a permutation of corresponding constituent characters leads to a key. PSO based cryptanalysis involves an optimal key search [1]. Each particle represents a 56-bit binary key string [2]. Initially, a particle is set to traverse the search space with randomly generated velocities. The intermediate velocity and the corresponding position of a particle is determined using the following formulae [16] [23]:

$$V_{in} = w \times V_{im} + X_i r_i (P_{im} - N_{im}) + X_j r_j (P_{im} - X_{im}) \ldots\ldots\ldots (4)$$

where m = 1,2,...,D, i=1,2,...,N, N is the size of the population, w is the particle weight, X_i and X_j are two positive constants r_i and r_j are two randomly selected values in the range [0,1]. P_{im} and P_{in} are the local and global best positions of the particle i [23]. With this algorithm and a key length of 56 bits, there are 2^{56} (approximately7.2×10 16 keys) possible keys. Initially, executing a brute-force attack seems impracticable with alphabet size of 26! characters [16]. However, as mentioned in Table 1, encryption process does not alter the language statistics and hence frequency analysis presents a powerful and relatively easier tool for breaking classical ciphers [21]. The proposed work attacks a Four-Round DES algorithm composed of substitutions and transposition ciphers [1]. The process is initialized with a group of random particles that serve as keys. The i^{th} particle in the search space is represented by its position in an N-dimensional space [12].

The two important parameters calculated by the PSO algorithm are Probabilistic transition rule and velocity updating [15].

Probabilistic transition rule: The probabilistic transition rule also known as random proportional transition rule is as follows [15]:

$$Pij = [\eta_{ij}]^{\alpha} . [\tau_{ij}(t)]^{\beta} / \sum\sum \eta_{ij} * \tau_{ij}(t) \ldots\ldots\ldots\ldots\ldots (3)$$

where Pij is the probability of particle being present at a point in the search space, η_{ij} is the heuristic value, τ_{ij} is the number of particles at iteration t, α is the total number of attributes, β is the number of domain values of an attribute and i and j are two adjustable parameters to control the weights of the heuristic and particle values respectively. The heuristic information indicates how often a particular best location has been chosen by different particles of the swarm. Therefore the value of the move probability depends on the result of the equation [15].

Particle velocity updating: Particle velocity updating is the process according to which the particle velocities are increased adaptively for all the particles in the swarm.. Furthermore, the probability to find best position in the space at point i fluctuates randomly. By normalizing the number of particles in the iteration, this objective would be satisfied [15].

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) * R \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

This work aims at studying the performance of cryptanalysis of four-round DES using PSO optimization method. This implementation is based on a fuzzy IF-THEN rule base to increase the interpretability and accuracy of cryptanalysis model. The values of parameters assumed in this paper for PSO include total no. of cycles = 1000, 5000 and 10000; no. of particles = 500, 1000, and 5000; maximum iterations =250, 500, 1000 and 9,000; α = 1 and β = 0.01, 0.5 and 1. The maximum velocity of the algorithm was set to 10. The size of population was taken equal to N = 10, 20 and 40, the number of iterations = 100and the number of Trials T=4.

The experiments were carried out in MATLAB SIMULINK environment using PSO toolbox and Fuzzy logic toolbox. The algorithm has been implemented successfully for varying strength of the cipher text. Table 2 represents the performance and simulation results of cryptanalysis using PSO for four-round DES.

ISSN: 2278 – 1323

Ciphertext size	Recovered bits from original key				Average number of success bits after four trials
	T1	**T2**	**T3**	**T4**	
4	31	27	36	33	35
16	32	34	38	35	36
64	39	37	42	37	33
256	41	39	41	36	39
1024	43	42	44	36	40

Table 1 Recovered key bits versus ciphertexts of varying size, for 4 trials

The Table 1 indicates that the size of ciphertext was varied and the average number of success bits that have recovered from real key are limited between 35 to 40 and the proposed algorithm has the ability to find out the complete key using N-gram or other language statistics.

A resulting original image, the encrypted image and the cipher image using 4-round DES is shown as follows:

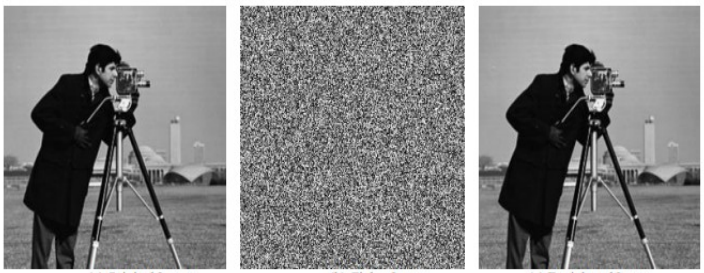

Figure 6 Image encryption using 4-round DES

VIII. CONCLUSION & FUTURE SCOPE

In this paper, Particle Swarm Optimization based cryptanalysis of four-round Data Encryption Standard is presented. Experimental results demonstrate this approach to be an effective means for cryptanalysis of four-round DES. The cost function and the fuzzy if then rules used in this paper are generic and can be applied to cryptanalysis of other block ciphers of any length. The convergence times for average of four trials distributed over various parameter values are comparatively less in view of the available computing power. As a future work, the authors look to implement this approach on five-round, eight -round and triple DES as well as on Advanced Encryption Standard algorithm. We will also try to improve the fitness function and use other variations of PSO. Apart from PSO, other CI paradigms such as Ant Colony Optimization and Artificial Immune Systems are also proposed to be used for cryptanalysis.

This paper concluded that Evolutionary Computation and Swarm Intelligence have been successfully applied to cryptology. Many cryptographic techniques have been developed and several were broken. Recently, new models based on the Computational Intelligence CI and bio-inspired techniques can be found in the literature showing their effectiveness in handling hard problems in the area of cryptology. However, some authors recognize that the advances have been slow and that more efforts are needed to take full advantage of CI techniques. In this work, we present a brief review of some of the relevant works in this area. The main objective is to better understand the advantages of applying CI on cryptology in the search for new ways of improving computer security. However, it is necessary to notice that many of those implementations have been either to classical systems that have no real world application, or results obtained are difficult to reproduce and verify.

References

[1] Albassall A.M.B., Wahdan A.: Genetic Algorithm cryptanalysis of a Fiestal type block cipher. Proceedings of ICEEC '04, pp. 217-221 (2004).
[2] Ali Aydın Selçuk.: On Probability of Success in Linear and Differential Cryptanalysis. In J. Cryptology Vol. 21, pp. 131–147 (2008).
[3] B. Carter and T. Magoc.: Classical Ciphers and Cryptanalysis. Technical Report (2007).
[4] Bárbara E. Sánchez Rinza, Diana Alejandra, Bigurra Zavala, Alonso Corona Chavez.: De-encryption of a text in spanish using probability and statistics. In proceedings of 18th International Conference on Electronics, Communications and Computers IEEE 2008, pp 75-77 (2008).
[5] Carlisle Adams.: Designing against a class of algebraic attacks on symmetric block ciphers. J. Applicable Algebra in Engineering, Communication and Computing Vol. 17, pp. 17–27 (2006).
[6] Eberhart, R.C. and Kennedy, J. (2001) Swarm Intelligence. London: Morgan Kauf-mann Publishers.
[7] Engelbrecht, A.P. (2007) Computational Intelligence: An Introduction. 2nd ed. Chichester : Wiley.
[8] K.W. Lee, C.E. Teh, Y.L. Tan.: Decrypting English Text using enhanced frequency Analysis. In proceedings of National Seminar on Science, Technology and Social Sciences 2006 pp. 1-7 (2006).
[9] Laskari, E. C., Meletiou, G. C., Stamation, Y. C., and Vrahatis, M. N., Evolutionary Computation based Cryptanalysis: A first study. Nonlinear Analysis, vol. 63, no.(5- 7), pp. 823-830, 2005.
[10] M.S.V.S. Bhadri Raju, Effect of Language complexity on Deciphering Substitution Ciphers - A case study on Telugu. International Journal of Security and its applications (IJSIA), Vol. 4, Issue 1, Science and Engineering Research Society (SERSC), Korea, pp. 11-20 (2010).
[11] Michael J. Wiener.: The Full Cost of Cryptanalytic Attacks. J. Cryptology Vol. 17, pp 105–124 (2004).
[12] R, Vimalathithan., and Valarmathi, M. L., Cryptanalysis of S-DES using Genetic Algorithm". International Journal of Recent Trends in Engineering, vol. 2, no. 4, pp.76-79, Nov. 2009.
[13] Ruth M. Davis, The Data Encryption Standard, Proceedings of Conference on Computer Security and the Data Encryption Standard, National Bureau of Standards, Gaithersburg, MD, Feb. 15, 1977, NBS Special Publication 500-27, pp. 5-9.
[14] Sean Simmons, Algebraic Cryptanalysis of Simplified AES. In J. Cryptologia, Vol. 33, pp 305–314 (2009).
[15] Seung-Jo Han, The Improved Data Encryption Standard (DES) Algorithm, pp. 1310-1314 (1996).
[16] Shahzad, W., Siddiqui, A. B., and Khan, F. A., Cryptanalysis of Four-Round DES using Binary Particle Swarm Optimization. Genetic and Evolutionary Computation Conference, pp. 1757-1758, July 8-12, (2009).
[17] Song, J., Zhang, H., Meng, Q., and Wang, Z., Cryptanalysis of Four-Round DES Based on Genetic Algorithm. International

ISSN: 2278 – 1323

Conference on Wireless Communications Networking and Mobile Computing, Issue 21-25, pp. 2326-2329. (2007).
[18] Stallings, W. Cryptography and Network Security Principles and Practices. Pearson Education, (2004).
[19] Subbarao V. Wunnava, Data Encryption Performance and Evaluation Schemes, Proceedings IEEE Southeast conference, pp. 234-238. (2002)
[20] Sujith Ravi and Kevin Knight.: Attacking Decipherment Problems Optimally with Low-order N-gram Models. In proceedings of the conference on Empirical Methods in Natural Language Processing, pp. 812-819 (2009).
[21] Sujith Ravi, Kevin Knight.: Attacking Letter Substitution Ciphers with Integer Programming. In J. Cryptologia Vol. 33, Issue 4, pp. 321-334, (2009).
[22] Toemer, R., and Arumugam, S. Breaking Transposition Cipher with Genetic Algorithm. Electronics and Electrical Engineering. vol.7 no.79, pp.75 – 78. (2007).
[23] Uddin, M. F., and Youssef, A. M. Cryptanalysis of simple substitution cipher using Particle Swarm Optimization. IEEE Congress on Evolutionary Computation, pp. 677-680, 2010.
[24] Z. Lin and H. Wang, "Efficient image encryption using a chaos-based PWL memristor," IETE Technical Review, vol. 27, no. 4, pp. 318–325, 2010.

ISSN: 2278 – 1323