Dual CBC Encryption Algorithm

Omprakash Kar[#1], Manoranjan Panda[*2]

[#] M.Tech Student, Department of Computer Science and Engineering, College of Engineering and Technology, Bhubaneswar, Odisha, India

[*] Lecturer, Department of Computer Science and Engineering, College of Engineering and Technology, Bhubaneswar, Odisha, India

Abstract: *Now-a-days we are enveloped by technologies and data is an essential part of any technology. It is obvious that communication between different clients is an important aspect of any system. Thus data are sent in between different clients to accomplish communication. After a plaintext is encrypted and cipher text is found, the cipher text is made to transmit over an unsecure channel. Encryption techniques are basically divided into two types: Symmetric key encryption and Asymmetric Key encryption. Symmetric key encryption is usually preferred over asymmetric key due to its fastness, low complexity, lesser computational burden and use of one key for both encryption and decryption process. Previously different encryption algorithms were developed i.e. DES, 3DES, AES and RC2 as symmetric key and Diffie-Hellman and RSA as asymmetric key algorithms. In this paper we have developed an encryption algorithm "Dual CBC Encryption Algorithm (DCEA)" which uses a twice implementation of Cipher Block Chaining (CBC) once in practical order and other in reverse order of the outputted cipher text. The use of XOR operation eases the hardware implementation, consumption of lesser computational power, smaller complexity and easy encryption and decryption processes. The key used here could be of any length but it is preferred to select a key between 4 Bytes and 16 Bytes. On comparison of Avalanche Effect with the other basis Algorithms like Blowfish, 3DES, DES, RC2 and AES algorithm DCEA has a much better value. Moreover DCEA is trustworthy to several attacks like frequency analysis attack, cipher text only attacks, known plaintext attack and chosen plaintext attack. In addition to that the algorithm has better encryption time in comparison to DCA encryption algorithm. The experimental and analytical results described in further sections of this paper give stronger proof to DCEA's strength.*

Keywords: *Avalanche Effect, Chosen Plaintext Attack, Cipher Text Only Attack, Encryption Time, Frequency Analysis Attack, Known Plaintext Attack, Symmetric Key Algorithm, Throughput*

I. INTRODUCTION

The world is edging towards a rapid development in the field of technology. Thus we can fluently announce that we are surrounded with technologies. So there are lot many areas where these technological applications are used such as social networking, ATM transaction, banking, stock exchange etc. So communication in between different systems is an important aspect of mingled technologies in which different clients tend to exchange data or information amongst themselves. So the data which is sent (mostly over in an unsecure media) in any wired or wireless media needs to be protected as such confidential transactions over wire or wireless public networks demand end-to-end secure connections to ensure data authentication, privacy and integrity [2]. For this we use different kind of encryption algorithms to secure the data and ease a fruitful communication.

A. Vital terms used in cryptography

Plaintext: - It is the text that a person choices to send to another person. For example, if a person A wants to send a text "Hello" to person B then that text "Hello" is the plaintext for the system.

Cipher text: - This is the text that the original message is transformed into to be sent over any public media. This is the encrypted text which is processed after the encryption of plain text. For example, if person A sends a text "Hello" to another person B, it is refined into some text like "As09/2?d57" which is called the cipher text.

Encryption: - A process of converting Plain Text into Cipher Text is called as Encryption [3]. Usually a data is sent over an insecure channel after encryption, so the confidential messages are processed using an encryption algorithm to be sent over any insecure wired or wireless media. For the encryption process requires a key and encryption algorithm, which is done in the sender's side.

Decryption: - A process of converting a cipher text back into plain text is called Decryption. The decryption algorithm along with key is used at the receiver side to achieve back the plain text. Hence Decryption is just the inverse of encryption process.

Key: - A Key is a numeric or alpha numeric text or may be a special symbol [3]. In the case of encryption it is used in the sender's side and in the case of decryption it is used in the receiver's side. In most of the algorithms the length of key decides the strength of the encryption.

B. Objectives of Cryptography

Confidentiality: - Information transmitted from a person should be addressed to the authorised receiver and not to someone else.

Authentication: - It checks the identity of the sender, if the received information from authorised party or from any fake identity.

Integrity: - No one instead of the authorised person is allowed to modify the file/text sent over the network to the receiver by the sender.

Non Repudiation: - None either the sender or the receiver is allowed to deny the transmission of data.

Access Control: - Only the authorised person (receiver or sender) is used to access or achieve the presented information.

Now to ensure the security and efficiency of a communication a variety of encryption algorithms are used. All these algorithms are divided into 2 types: Symmetric key encryption which is also called secret key encryption(which uses a secret key for both encryption and decryption) and Asymmetric key encryption which is also called as public key encryption(which uses a public and a secret key). The symmetric key algorithms are excessively fast and are little complex, so this eases the implementation on hardware. On the other hand asymmetric key algorithms are awfully slow (around 10^3 times sluggish) and extensively complex. Moreover Symmetric key algorithms are used to send large packets of data but Asymmetric key algorithms just enjoy the right to exchange secret keys. This gives a clear vision of the reason behind the use of symmetric key encryption algorithm in DCEA. Finally the symmetric key encryption techniques are used in algorithms like DES, 3DES, AES and RC4 and asymmetric key encryption techniques are used in algorithms like Diffie-Hellman and RSA.

Usually the asymmetric key algorithms are used along symmetric algorithms to give better performance so a huge size key is used in the case of asymmetric key algorithm, for which asymmetrically ciphered text are tenacious to decrypt back. But this is not valid proof that the asymmetric key encryption algorithms are better than symmetric key encryption algorithms as this leads to some other problems like computational burden(CPU time, execution time, burst time, throughput) and ease of distribution[1].

A key is a numeric or alpha-numeric text or may be a special symbol [1]. The key is used at the time of encryption takes place on a plaintext and at the time of decryption takes place on a cipher text [1] [3]. In cryptography the choice of key is vital as it determines the strength of ciphered text, basically a key size of 8-32 Bytes is preferred in most of the encryption algorithms. The strength of the encryption algorithm relies on the secrecy of the key, length of the key, the initialization vector, and how they all work together [1][4]. Moreover in the case of public algorithms (where the details of algorithms are known in the public domain) it is advised that the key should be of a large length to decide the security of the encrypted text as it would massively confuse attackers to get to the plain text and the key.

Now shifting our focus onto the secrecy of the system C. E. Shannon in the year 1949 came up with some benchmarks, which proposed amount of secrecy, size of key, complexity of Enciphering and Deciphering Operations, Propagation of Errors and expansion of message [6]. This also leads to the examination of the quality of a system taking into account the confusion and diffusion principles [1]. Again an acceptable cryptographic system should be resistant to a variety of attacks i.e. Linear Cryptanalysis, Differential Cryptanalysis, The Boomerang Attack, algebraic attacks etc [7].

Going to the background study to determine the literature survey of this paper, so in [1] a new encryption algorithm called Double Chaining Algorithm (DCA) is proposed by Daniar Heri Kurniawan and Rinaldi Munir which is modified in this paper later to exploit the relative performance of DCA and DCEA with performance and newer security criteria. There is a deeper analysis of two popular algorithms DES and Blowfish algorithms on the basis of performance namely power consumption and encryption speed in [2].A rooted scrutiny on DES, RSA and AES is undertaken taking into considerations of Stimulation Speed, Hardware & software implementation, ciphering & deciphering algorithm, encryption and decryption time in [8]. The research paper [3] promotes image encryption which guards upon image encryption, video encryption, and chaos based encryption which has applications in many fields including the internet communication, multimedia systems, medical imaging, tele-medicine and military communication, etc. The encryption and decryption time of skipjack and blowfish algorithms are described in [4]. Paper [5] demonstrates a new cryptographic algorithm for real time system having better benchmarks on encryption and decryption time.[9] and [11] analyses about different cryptographic demonstrating encryption time, CPU process time, CPU clock cycles, throughput and battery power in [9] where throughput of encryption and decryption of AES, DES, 3DES, Blowfish for ECB and CBC in [11]. [12] and [13] differentiates AES, Blowfish and DES, RSA respectively. A comparison of all algorithms DES, 3DES, RC2, RC6, Blowfish and AES is concluded in [14] using a variety of yardsticks. Various sorts of genetic algorithms and their applications are described in [15].

In this paper we have proposed an algorithm called "**Dual CBC Encryption Algorithm (DCEA)**" which implements the cipher chain blocking (CBC) algorithm twice, once in the logical order and other in the reverse order of the generated cipher text from first phase using the same initialisation vector (IV) in both the occasions. Moreover we do not directly use a key but we encrypt it using MD5 hash algorithm and find sub keys which is discussed in the later sections of this paper. Henceforth the encryption is carried on in two phases of encryption stages. Later a substitution box implementation is carried on the merged texts after plaintext is processed in encryption phases.

This algorithm is further scrutinized against different attacks like frequency analysis attacks, chosen plain text attack, known plain text attack and cipher text attack to guard upon the security analysis of the system. The performance is analysed using the Avalanche Effect of DCEA with other existing algorithms like Blowfish, 3DES, AES, DES, RC2. Again the encryption time and throughput of this algorithm is examined against Double Chaining Algorithm (DCA) as proposed in [1]. The detailed explanation of this encryption algorithm is presented later in this paper.

II. SKETCH OF EXISTING ALGORITHMS

The process of encryption is to convert a plaintext to ciphered text for which an encryption algorithm along with a key is used to transform a plaintext to ciphered text. Encryption could be either based on stream ciphers or block ciphers. The length of key, type of key (only alphabets, alphabets with numbers, alphabets with numbers and spaces or alphabets with numbers, spaces and special characters) determine the strength of the encrypted text. In most of the public encryption algorithms (where algorithms are publicly declared) it is advised to use a stronger key to test the quality of encrypted text. For symmetric key algorithms only a particular key is used for both encryption and decryption process but in the case of asymmetric key algorithms different keys are used for both encryption and decryption process. There are different encryption algorithms used, some of them are exampled below.

DES: - DES stands for Data Encryption Standard which is an implementation of Fiestel Cipher. It uses a symmetric-key block cipher released by National Institute of Standards and Technology (NIST). It uses 16 rounds having a block size of 64 bit and key size of 64 bits. But the fact is that it only uses 56 bit key and the rest 8 bits are used as redundant bits. This uses steps like initial permutation, 16 fiestal rounds left right swap and final permutation.

Triple DES: - This is the expansion DES which consists of 3 different keys K_1, K2, K3. So the length of key is tripled to 168 (3 x 56). The single DES could be implemented thrice with the same key find the same result as 3DES but 3DES uses 3 different keys which is not possible with DES. Triple DES is simply more secure than DES but the encryption time is much higher than DES.

AES / Rijndael: - AES stands for Advanced Encryption Standard which uses symmetric key symmetric block cipher. It uses 128 bit data and a variety of key lengths 128 bits, 192 bits, 256 bits. In comparison to DES and triple DES, AES is much stronger and faster. It is an iterative model instead of feistel model which uses a substitution and permutation model. Substitution depicts that a particular item is replaced by some outputs and permutation suggests the shuffling of bits around. The encryption process describes different stages of shift rows, mix columns and add round key.

RSA:- Rivest-Shamir-Adleman (RSA) is a public key cryptography algorithm which was named after Ron Rivest, Adi Shamir and Len Adleman, and invented in the year 1977. RSA is used for public key encryption and digital signatures and security is established on the difficulty of factoring a product of two large numbers. It is relatively slower algorithm as compared to other cryptographic algorithms and has limited use to encrypt data.

RC2:- The RC in RC2 stands for Ron's code which was designed by Ron Rivest in the year 1987. This is also called as ARC2.RC2 uses a Feistel network in which 18 rounds are arranged as a source heavy manner. It uses 16 rounds of mixing punctuated by two rounds of mashing. The whole process consists of a 64 bit block cipher and a variable length key.

RC6:- It is a variant of RC6 which uses symmetric key block cipher derived from RC5. It uses a block size of 128 bits and supports a variety of key sizes i.e. 128, 192, and 256 bits up to 2040-bits. A mixture of parameters could be put into the algorithm to support large selections of word-lengths, key sizes, and number of rounds. RC6 uses data-dependent rotations, moduar addition and XOR operations.

Blowfish: - Blowfish encryption algorithm is a symmetric block cipher which uses a variable-length key from 32 bits to 448 bits which is optimal for both domestic and exportable use but till date it is not implemented in practical use. It uses a 64 bit block cipher using a key ranging from 32 bits to 448 bits. It was developed by Bruce Schneier in the year 1993 making it faster against DES and IDES. Moreover

Blowfish does not require any license and is unpatented and royalty-free.

III. COMPONENTS OF A SECURE COMMUNICATION

In the year 1949 C.E Shannon came off with some appraisements to judge the secrecy of a system through the research paper “communication theory of secrecy of system” [6]. Here he suggested some guidelines for the best secret system. Moreover some basic ideas of good ciphers and methods are described in this paper. So the followings are the benchmarks to an acceptable cipher:

Amount of Secrecy: – The basics of a cryptosystem are to accept a plaintext and encrypt it into a cipher text and send it into an unsecure public network. So the attackers always have an intention to manipulate the encrypted text and get back to the plaintext or the algorithm which is used by the cryptosystem. In cryptography the basic idea is to use an efficient algorithm so that the algorithms or calculations used in the algorithm could not be decrypted back. Finally the secrecy of cryptosystem decides the amount of hidden processes proceeding in the system.

Size of key: - There has been a debate on how much length a key should be. Some say that the key must be transmitted from the sending point till the receiving end or else some other time the key is to be remembered, hence the key should be kept as small as possible. On the other hand according to Kerckhoff’s principle if any algorithm is public (the algorithm is publicly known) then the key should have high strength, which determines the quality of ciphered text.

Complexity of Enciphering and Deciphering Operations: - It always advised to have algorithms with higher complex functions. But sometimes higher complexity leads to larger throughput and encryption time. But it is always beneficial to have permissible complexity in any algorithm to maintain the secrecy of any system. If the encryption and decryption operation are done manually it leads to loss of time and errors.

Propagation of Errors: - In fewer cases we find that during the encryption process a change any bit in transmission or substitution leads to a massive change in the ciphered text. This leads to massive loss of information and virtually a different decrypted text. Finally it is always advisable to have the least propagation of errors.

Expansion of Message: - Many encrypting systems have the tendency to expand a message may be due to double substitution over expansion of message. A larger text means a higher size of file has to move through the network which might increase the load of any system.

A. Substitution-Permutation Network

Claude Shannon in the same research paper [6] came up with another concept of S-P Networks (Substitution-Permutation Networks) which describes the linked mathematical operations of block ciphers algorithms such as AES, PRESENT, SAFER, SHARK, and Square. The networks usually take a block of text and a key which are functioned, transformed and altered to find the required cipher block. The S-box and P-box are used to act upon the sub blocks of the function to find the required final output. These are used for better implementation on hardware by using XOR and bitwise rotation.

S-box: - A S-box substitutes 1 block of bit to another block of bit one by one in order to ensure decryption. In general the size of input block should be same as the size of output block (if input size of block is 16 bytes then the output should be of 16 bytes) but is variant in case of DES. A satisfying S-box should have the criteria that changing one bit of input would change more than 50% of characters in encrypted text.

P-box: - The output of all substitution boxes are taken and fed back into the s-box after permutation by the P-box. It ensures that a particular s-box is evenly distributed to different s-boxes.

Finally the s-box could be thought of as a substitution cipher and p-box could be considered as a transposition cipher. Now a well constructed S-P network with changing or altering rounds of S-Boxes or P-Boxes satisfies Shannon’s confusion and diffusion benchmarks:

Diffusion: - If one bit is altered in the plaintext then all these effect the relative S-box and P-box operations resulting in changing the output to several bits. According to strict avalanche criteria the output bits should be changed by more than half as compared to that of the inputted text. So lastly it can be concluded that we take a plaintext with a particular key, find its cipher text and again flip one bit in the plain text and find the cipher text which results in the changing of several bits of both ciphered text.

Confusion: - It states that changing one bit of key changes different round keys. This results in a complex encrypted text as a particular key diffuses over a particular round.

B. Avalanche Effect

Considering the diffusion criteria into consideration a desirable property to cryptographic algorithms especially ciphers and cryptographic hash functions. It states that if a bit is slightly flipped in plaintext the more half of bits are flipped in cipher text. Although this term was first coined by Horst Feistel by the idea was first developed by C.E Shannon. SHA1 shows best results against avalanche effect as it flips the cipher text completely [1]. The equation to avalanche effect is:

$$\boldsymbol{Avalanche\ Effect} = \frac{\boldsymbol{No.\ of\ flipped\ bits\ in\ cipher\ text}}{\boldsymbol{No.\ of\ bits\ in\ cipher\ text}} \quad (1)$$

Eqn. 1 Equation of Avalanche Effect

IV. CRYPTANALYTIC ATTACKS ON SYSTEMS

We came across several kinds of cryptographic algorithms such as DES, Triple DES, RSA, RC2, RC6, Blowfish etc. Now we are going to analyse the attacks to the cryptosystems. Almost all the cryptographic algorithms are designed in such a manner that they counter these attacks. But the attackers find different impish ideas to sense what’s happening in a cryptosystem internally. These attacks use mathematical weaknesses of algorithms or attack is made on specific part of implementation of the algorithms. Some of these attacks are sited below.

Known-Plaintext Attack: **-** In this kind of attack the assailant/attacker has the idea of both plaintext or crib and its encrypted or cipher text. These clues of plaintext and cipher text are used to explain other secret information like key or code books.

Cipher text only Attack: **-** The attacker has a collection of several cipher texts. The attacker tries to decode the cipher text to find the key or find their respective plaintexts.

Chosen Plaintext Attack: **-** Here the attacker pre guesses that a particular plain text that can be retrieved from the corresponding cipher text.

Chosen Cipher Text Attack: **-** In the Chosen Cipher text attack, the attacker has the intelligence to recapture the plaintext from a cipher text and inspect them.

Chosen Text Attack: **-** The attacker with the ability of both chosen plaintext attack and chosen cipher text attack come into the chosen text attack.

An encryption algorithm is said to be unconditionally secure if the attacker does not have any idea how to access back the plain text from the cipher text, regardless of how much access to the cipher text the attacker might be having. An algorithm is said to computationally secure if the cost of cracking the cipher text exceeds the cost of encrypting the text and the time to decrypt the encrypted text exceeds the useful lifetime of the system.

V. PRIMITIVES USED IN ALGORITHM DESIGN

MD5: - MD (message digest) is a collection of several cryptographic algorithms MD2, MD4, MD5, MD6. MD5 encryption algorithm results in an output of 128 bit hash value resulting in a hexadecimal string of 32 blocks (as a hexadecimal bit constitutes of 4 bits). The benefit of a hash algorithm is that these allow a one way encryption operation which cannot be decrypted back. It is suitable in the case of checksum to verify data integrity.

XOR (Exclusive or Operation): - The XOR operation outputs true if one input is true and other is false. It results false if both inputs are true or both inputs are false. XOR has several advantages in cryptography i.e. it is very fast computable and can be easily implemented on hardware, it can be easily encrypted or decrypted, the commutative and associative laws are easy to compute and XOR is easy to analyse and understand.

CBC: - CBC (Cipher Block Chaining) is a block cipher mode of encryption which is used for information security services such as confidentiality and authenticity. The algorithm first divides an inputted text into blocks of equal sizes. This uses an Initialisation Vector (IV) which has an XOR operation with the first block to find a cipher text. Again the resulting cipher text has the XOR operation with the second block to find the second cipher text. This process is continued till the last block of the inputted text series to find the cipher text blocks.

Single Point Crossover Genetic Algorithm: - In this scenario a crossover point is selected and the resulting string is created by taking binary string from first till the crossover point from first parent and rest from the second parent or vice versa.

VI. DUAL CBC ENCRYPTION ALGORITHM (DCEA)

The Dual CBC Encryption Algorithm (DCEA) implements the Cipher Block Chaining (CBC) algorithm twice, once in the practical order and other in the reverse order of the resulting cipher text. It can be summarised that DCEA uses a double CBC encryption method to carry out its operational mechanism. Now explaining the idea behind CBC algorithm:

A. Cipher Block Chaining (CBC)

This uses a block cipher mode encryption method in which blocks of text are encrypted with a particular key applied to the whole block. In addition to this CBC uses an Initialisation Vector (IV) of any finite length which is used to initiate the encryption process. In this algorithm the initialisation vector has an XOR operation with the first text block to find the first cipher text. Again the first cipher text has an XOR operation with second text block(encrypted blocks after encrypting first text blocks) in reverse order to find the second cipher text. This process continues till the last text block to find a collection of cipher texts. Fig. 1 explains the CBC algorithm:

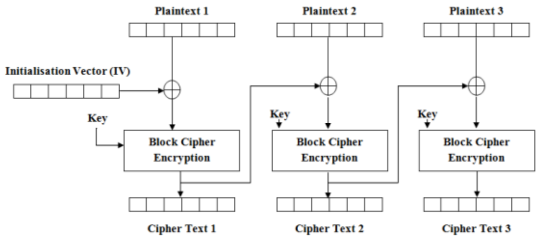


Fig. 1 CBC Algorithm

B. *Dual CBC Encryption Algorithm (DCEA)*

The DCEA algorithm accepts an infinite length string and divides them into plaintext blocks of 16 byte each. If the length of the last block is less than 16, then the string is padded up to 16 bytes with the symbol *. Then it uses the CBC algorithm twice, once in the incremental order to find the cipher text blocks and again in the reverse order of the resulting cipher text blocks using the same initialisation vector in both the occasions. So for example if we have 5 plaintext blocks we have 5 cipher text blocks. Moreover this algorithm doesn't use a key directly in the cryptosystem rather a set of 12 sub keys to carry on with the encryption process. Fig. 2 sites the explanation of double CBC mode.

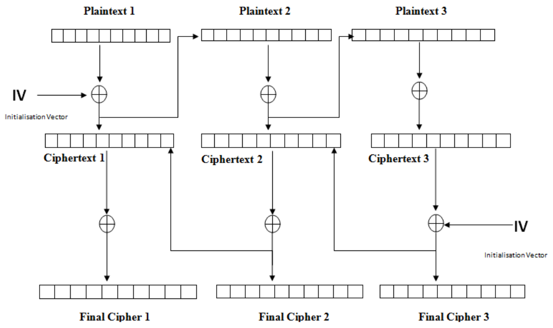


Fig. 2 Double CBC Algorithm

C. *Sub Key Generation*

A key of any arbitrary length (preferably in between 4 bytes to 16 bytes) is selected. The key is liable to contain any alpha numeric digit with spaces. The following steps are followed to generate sub keys:

Encrypting key to MD5 hash: - The chosen key is first encrypted with MD5 hash function to find a 128 bit or 16 byte hash. This results in a hexadecimal output consisting of 32 hexadecimal symbols (as each hexadecimal symbol constitute of 4 bits).

Splitting The Encrypted Text Into 4 Equal Sections: - Now the hashed key after MD5 hashing (16 Bytes) is divided into 4 parts each of 4 bytes (D1, D2, D3, D4).

XOR Operation: - The 4 digests of key have a XOR operation ($D1 \oplus D2$ and $D3 \oplus D4$) to find the resulting output of two blocks each of 4 bytes (X1 and X2).

First 4 Sub keys: - 4 Sub keys are generated from the XORed X1 and X2 by splitting X1 and X2 into each of 2 Bytes.

Single point crossover to find rest 8 sub keys: - The idea of single point crossover is used on the 4 sub keys to find the resulting 8 sub keys. Two sub keys are taken in an incremental order to find sub keys. Fig. 3 and Fig. 4 depict the implementation of sub key generation.

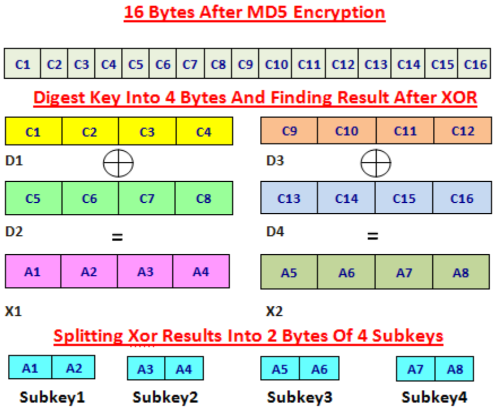

16 Bytes After MD5 Encryption
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16
Digest Key Into 4 Bytes And Finding Result After XOR
C1 C2 C3 C4
D1
C5 C6 C7 C8
D2
=
A1 A2 A3 A4
X1
C9 C10 C11 C12
D3
C13 C14 C15 C16
D4
=
A5 A6 A7 A8
X2
Splitting Xor Results Into 2 Bytes Of 4 Subkeys
A1 A2
Subkey1
A3 A4
Subkey2
A5 A6
Subkey3
A7 A8
Subkey4

Fig. 3 Finding First 4 sub keys

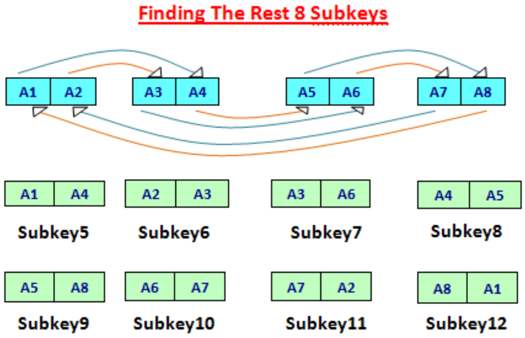


Fig. 4 Generating rest 8 sub keys using single point crossover genetic algorithm

D. *Encryption Process*

The encryption process consists of 2 phases chaining and substitution. As of now we are having finite number of plaintext blocks each of 16 bytes and 12 sub keys. The encryption process is carried on using the sub keys onto the plaintext blocks. Both chaining and substitution phases are discussed below.

Chaining Phase: - In the chaining phase plaintext blocks along with a particular sub key is processed with the CBC encryption mode. The chaining phase constitutes of 4 stages and each stage comprises of 2 rounds. At a particular time a single block is accepted by the system and iterated 4 times of CBC operation. In a stage the first byte of sub key is used as the initialisation vector for first round and has CBC operation over each byte of plaintext to find first cipher text. Again for round two second byte of sub key is used as initialisation vector and CBC mode is carried on with the first cipher text to find second cipher text. This process is carried on for four times. The resultant output after 4 stages is reversed and processed again into the system to find final cipher. The chaining stage is explained in Fig. 5.

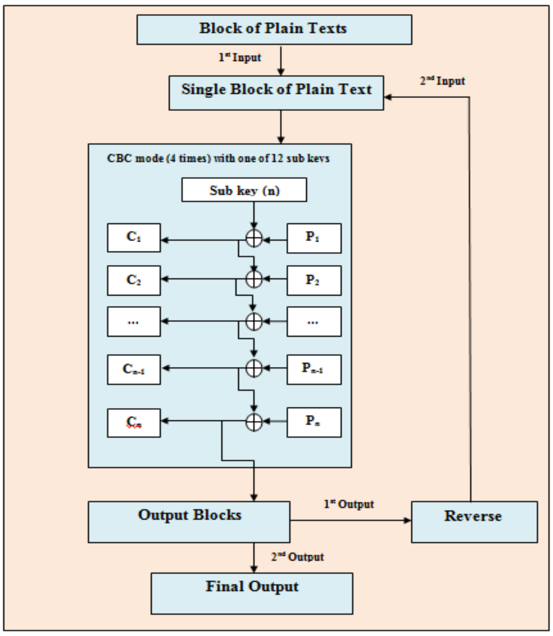
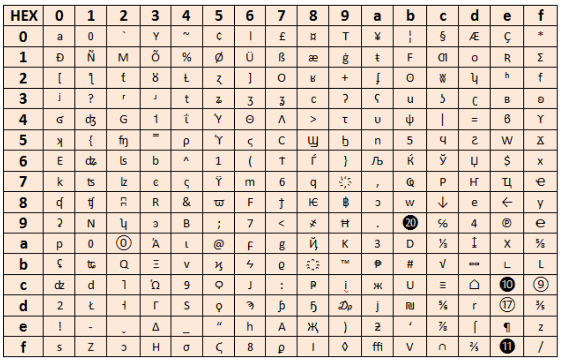


Fig. 5 Chaining process

Substitution Phase: - The output of the chaining stage is replaced by symbols from the S-box as described in Fig. 6. This process is carried on by spanning two hexadecimal characters and replacing them with a character from the S-box. The S-box consists of 256 symbols in a 16 X 16 grid constituting of 16 hexadecimal digits both in x-axis and y-axis. Hence the final cipher of the cryptographic algorithm is produced.



HEX	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	a	0	`	Y	~	¢	I	£	¤	T	¥	¦	§	Æ	Ç	*
1	Đ	Ñ	M	Ő	%	Ø	Ü	ß	æ	ġ	ŧ	F	Ɑ	o	Ʀ	Σ
2	[	ʅ	f	ɤ	Ł	ʐ	]	O	ʁ	+	ʆ	ʘ	₩	ɥ	ʰ	f
3	ʲ	?	ʳ	ʴ	t	ʑ	ʒ	ʓ	c	ʔ	ʕ	u	ʖ	ʗ	ʙ	ɵ
4	ɠ	ʤ	G	ɿ	ƭ	Ύ	Θ	Λ	>	τ	υ	ψ	|	=	ϐ	ϒ
5	ʞ	{	ʩ	⁼	ρ	ϓ	ς	C	щ	ђ	n	5	ч	ƨ	W	Ѫ
6	E	ʥ	ʪ	b	^	1	(	ϯ	ѓ	}	Љ	Ќ	ў	џ	$	x
7	k	ʦ	ʫ	є	ç	Ÿ	m	6	q	҉	,	Ҩ	Р	Ң	Ц	ɇ
8	ɖ	ʧ	ᴨ	R	&	ϖ	F	†	Ѥ	฿	ɔ	w	↓	e	←	y
9	ɂ	N	ɥ	э	B	;	7	<	ӿ	Ħ	.	⓴	%	4	℗	ɘ
a	p	0	⓪	Ά	ι	@	ϝ	g	Й	К	3	D	⅓	↨	X	⅝
b	ʕ	ʨ	Q	Ξ	v	ӽ	ϟ	ϱ	◌	™	₱	#	√	∞	ʟ	L
c	ʣ	d	˥	Ώ	9	Ϙ	J	:	Ᵽ	ị	ж	U	≡	⌂	❿	⑨
d	2	ł	ɟ	Γ	S	ϙ	ঌ	ƥ	ҕ	₯	j	₪	⅚	r	⑰	⅗
e	!	-	˯	Δ	_	“	h	A	Җ	)	ƻ	ʻ	⅞	⌠	¶	z
f	s	Z	ɔ	H	σ	ζ	8	ϼ	I	◊	ﬃ	V	∩	⅖	⓫	/

Fig. 6 S-Box

E. *Decryption Process*

The decryption process is achieved just in the reverse order of the encryption process. The symbols from S-box are substituted to the final cipher to find the hexadecimal number after chaining process. Again the reverse process of chaining phase is accomplished along with the sub key to find the respective plaintext.

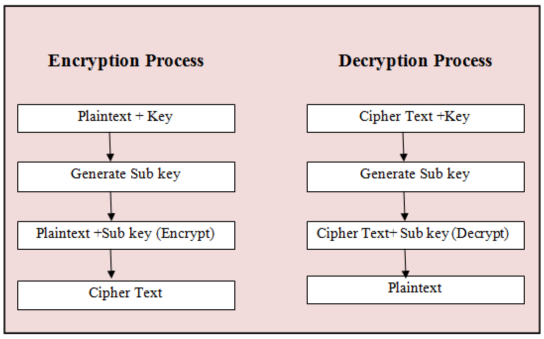


Fig. 7 Encryption and Decryption process

VII. EXPERIMENTAL RESULTS

The DCEA algorithm was programmed in java 7 using the IDE Netbeans 7.3.1. There is no specific reason to implement the algorithm in java platform as it could be implemented in other languages too. The test platform is a laptop, HP 6530s having specification of Intel core 2 duo processor (2.0GHz, 2 MBL2 cache, 800MHz FSB) and RAM of 2GB. The operating system used in this system is Windows 7.

A. *Performance Analysis*

The performance analysis of Dual CBC Encryption Algorithm (DCEA) is analysed on the basis of avalanche effect, encryption time and execution time throughput.

- ***Avalanche Effect***

The avalanche effect of DCEA algorithm was tested using same key and two plaintexts where the second plaintext in changed by one bit. In Table 1 a particular key is taken and encrypted with plaintexts where the second plaintext is just flipped by one bit in comparison to first one. In this table the first bit of plaintext 2 is changed in resemblance to plaintext 1.

Table 1 Avalanche Effect using same key

Key	Plaintext	Cipher Text
abcd	The term Odisha is derived from the ancient Prakrit word Odda Visaya	Ñƽ:ʰɔ(↓ÑƳ⓴ἴ{⑰ʒjϐϚϱ *ġ_.ʅ⓫w฿≡ʪϦƚ∞ȵΘZ_ ◊ψ>-VnʒF[fȵ9kþḰ@fŋʪ ȹʧɘK⓫_ fŋʫʣﬃʁ¢ʳɜʤ⑨ #Θɞ¢A7nˈBOɔ¶bg
abcd	Hhe term Odisha is derived from the ancient Prakrit word Odda Visaya	Σi|“£ʞɔ჻ʻ჻fŋʔ⌑Цψ=ϼŧH{ ⓴ꞪʨʨHASʢЦTϐ£ιψЖß Õt=Ħ pЦ⅝%ϐ◊Õw™ℨʧϟ ʹϒʕʫᎷʒʨVϐɨ&⅚.4tϐ+Цʡx[rƽY[჻⓫Жƽɨ! ч
Avalanche Effect		**1**

Now we take a different key and plaintext 1 to find the respective cipher text and analyse the avalanche effect of cipher texts 1 and 2 as represented in the Table 1. Table 2 states the study of avalanche effect with different key and plaintext 1 to the previously represented examples.

Table 2 Avalanche Effect using different key

Plaintext	Encryption Algorithms	Avalanche Effect
Technology Best.	Blowfish	0.2343
	3DES	0.1290
	AES	0.7530
	DES	0.1999
	RC2	0.9375
	DCEA	1

Again the avalanche effect of the above mentioned encryption algorithms are analysed taking file sizes in an incremental order of size of size of files. Fig. 8 represents a graph having size of bytes in x-axis and Avalanche effect in y-axis.

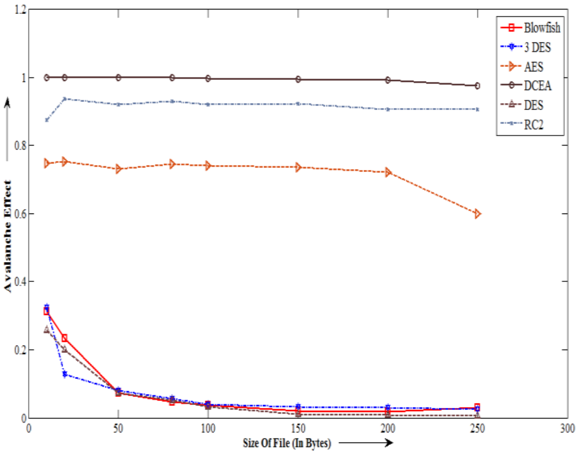


Fig. 8 Comparison of avalanche effect of different encryption algorithms

According to Shannon's diffusion criteria if the cipher texts differ more than half then the diffusion criteria is said to be fulfilled [6]. According to the graph represented in the Fig. 8 encryption algorithms like blowfish, 3DES, DES are par below the 0.5 in average which does not satisfy the diffusion criteria of Shannon. AES is marginally above the eligibility criteria and RC2 shows a good result to the diffusion principle. Finally DCEA shows the best results towards avalanche effect which is mostly above 0.97.

- ***Encryption Time***

In the research paper [1] a new encryption algorithm, DCA was developed and the DCEA is a variant of the DCA algorithm. In this section the encryption time of both the algorithms are calculated and a thorough analysis is carried on. Now file sizes of varying bytes are taken in incremental order and the encryption time is calculated. Table 4 gives the representation of various file input sizes (in bytes) and encryption time (micro sec). Fig. 9 gives a graphical representation of the whole scenario having file size in x-axis and encryption time in y-axis.

Table 4 Encryption Time of DCA and DCEA

Bytes	DCEA (in micro sec)	DCA (in micro sec)
12	12	12
120	14	21
240	25	26
360	27	54
480	30	78
600	59	91
720	63	187
840	104	305

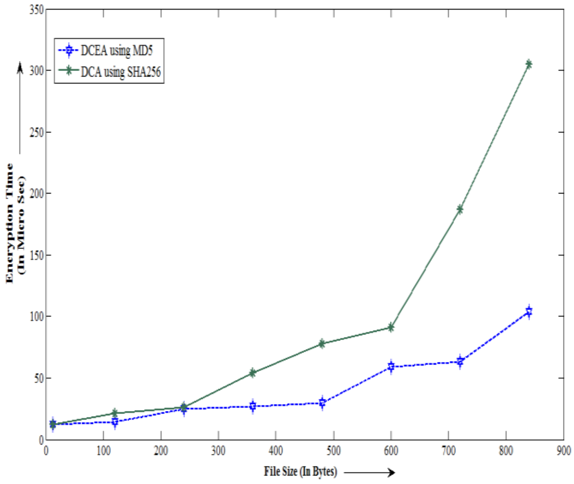


Fig. 9 Encryption time of DCA and DCEA graphically

In DCA algorithm we use SHA256 hash algorithm whereas in DCEA it uses the MD5 hash algorithm. So the number of bytes in sub key is reduced resulting in the further decrease of the rounds in the chaining phase. On an analysis to the graph represented in the Fig. 9 it is found that the DCA encryption graph shows an exponential growth of its curve in comparison to the DCEA encryption graph which shows a steady increase in its curve.

- ***Execution Time Throughput***

Throughput is the amount of material/ information passing through any system or process. The execution time throughput could be calculated in the following form:

$$Encryption\ Time\ Throughput = \frac{\Sigma\ Inputted\ \ Files}{\Sigma\ Execution\ \ Time} \qquad (2)$$

Eqn. 2 Equation of Encryption time throughput

Analysing the throughput of both DCA and DCEA:

<u>**DCA**</u>: -

Σ Inputted Files = 12 +120 + 240 + 360 + 480 + 600 + 720 + 840 = 3372 bytes

Σ Execution Time = 12 + 21 + 26 + 54 + 78 + 91 + 187 + 305 = 774 micro sec

Encryption time throughput = $\frac{\Sigma\ Inputted\ \ Files}{\Sigma\ Execution\ \ Time} = \frac{3372}{774}$ = 4.356 bytes/micro sec

<u>**DCEA**</u>: -

Σ Inputted Files = 12 +120 + 240 + 360 + 480 + 600 + 720 + 840 = 3372 bytes

Σ Execution Time = 12 + 14 + 25 + 27 + 30 + 59 + 63 + 104 = 334 micro sec

Encryption time throughput $\frac{\Sigma\ Inputted\ \ Files}{\Sigma\ Execution\ \ Time} = \frac{3372}{334} =$ 10.095 bytes/micro sec

Hence it can be studied that the encryption speed of DCEA is better than DCA.

B. Security Analysis

The security of DCEA is analysed on the basis of frequency analysis attack, known plaintext attack, cipher text only attack, chosen plaintext attack.

- ***Frequency Analysis Attack***

The DCEA algorithm uses a S-Box in the second phase of encryption phase, it might be an intuition that the algorithm might be prone to frequency analysis attack. The occurrence of each character in plaintext is found out and compared to the number of occurrence in cipher text. For a better encryption algorithm the frequency of characters in plaintext should be absolutely different to that of characters in cipher text. For example in Fig. 10 the frequency of each character in plaintext (English) is presented in the form of a histogram where it is found that o has a frequency of 44, f and n have frequency of 23, m has frequency of 16 and traces of other alphabets. Again in Fig. 11 the frequency of characters of cipher text is analysed where we find that the frequency is spread over several characters. Both Fig 10 and Fig 11 represent a histogram where no of characters is presented in x-axis and frequency is presented in y-axis.

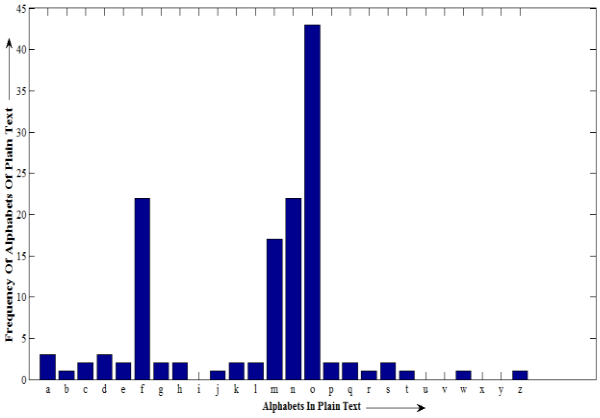


Fig. 10 Frequency of characters in plaintext

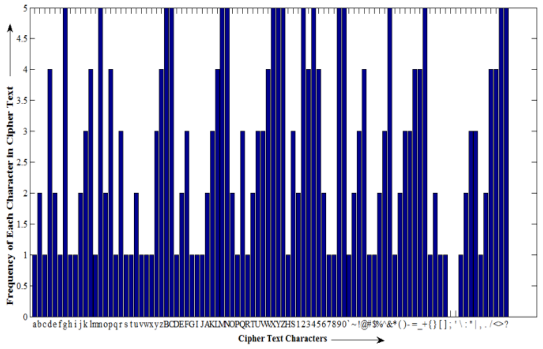


Fig. 11 Frequency of characters in cipher text

Again Table 5 analyses different cipher texts by encrypting separate plaintexts and the resulting cipher texts are analysed.

Table 5 Frequency Analysis of several plaintext and cipher text in DCEA algorithm

Sl No.	Plaintext	Cipher Text
1	oooooooooooooo ooooooooooo	Σʔnr¥ɬƢɭNҁRsc⅗ƔҔʙ0τM˯ƛȼȋɁŸ JɫρϴȡF!⅚p
2	nsnsnsnsnsnsnsns nsnsnsnsn	Ꞧρæ⅚pɔ⌂ʑʪQPKz`ʁƔΔʅe⅝E∩Δʆ ĐUc≡ꝶ{
3	bookbookbookbo okbookbook	vRHʅ~s{TA⅚ʦwl⓾ƔȸₚȣḰ$R<6f ʣ∩Ʀ⅗*1Ħɤfŋɔϔ
4	omprakashkarom prakashkar	ϴʣ|ʰ⓪ɔ¤ʤτ“₩₪%ΞZMQȸₚ^nҔs Ƕ$ʔ;(ҁSr℗ҔUʞ

- ***Known Plaintext and Cipher Text only Attack***

In the plaintext attack the attacker has a collection of both plaintext and cipher text where as in the case of cipher text only attacks the assailant has several cipher texts and tries to decrypt them to their respective plaintext. To have a check on these attacks the encryption algorithm produces different cipher texts from same key and same plaintext. This idea is accomplished by taking sub keys randomly from a set of 12 sub keys. So the probability of uniquely matching plaintext and cipher text is reduced to 0.084 which was previously 1 for several encryption algorithms. For example if a company wants to send a particular message to several employees using a single key at a particular time, it is always beneficial to send encrypt the text to different cipher texts to create a bewilderment in the minds of the attacker. Table 6 describes that the same plaintext and same key is encrypted for 10 times to find respective cipher texts.

Table 6 Same key and plaintext with different cipher text

Key	Plaintext	Cipher Text
abcd	CET BBSR	<⅚̮̑̑̑8$ꞓ€⅞`$DȣΩ!℗ʴɐ∩
		ȼ⅚⓴UrԿ?Z§N1sɬqAɓ⅝
		₩ɔ:≯ʖoɵЉGÜɐʮ⌂™k
		ȼ⅚⓴UrԿ?Z§N1sɬqAɓ⅝
		ʋ⓴√aɰ⅞æ55ʣfʦ¤Σϐɘ
		Rɬ⓫[ʡɢ{xʨŸʄj(⅚ɥ[
		|ʃ@ΣжвFɞƘeʨßȿ`≡ʁ
		ƉIρɔʻ7Çжʞ%ɰ*ʘ⁼ɰ₪ȣϒʢ
		F≯bŸ$ϟ₩ʄUҴ>⑨ʴʐꝶLꝭȡ
		₩ʦƛ⓪ҔҴÜЄƔ?3ʮϔ¤@YO

- ***Chosen Plaintext Attack***

The attacker here puts different plaintexts into the cryptosystem and has some idea till some parts of the encryption process. Hence to develop confusion in the minds of the intruder the algorithm shuffles the byte order in the chaining process. In DCEA chaining process the first stage and forth stage use first and second byte of sub key as initialisation vectors to the rounds and in the second and third stages, second and first byte of sub key is used as initialisation vectors of the rounds.

VIII. FUTURE WORK

The DCEA algorithm is basically divided into 3 stages sub key generation, chaining stage and substitution phase. Now for each stage there can be some improvement. Firstly a new concept of generating more sub keys could be described so that the known plaintext and cipher text only attacks could be checked more sincerely. Again for every stage a sub key could be introduced to control chosen plaintext attack. Moreover the size of the cipher text could be further shortened to increase the firmness of the cipher text.

XI. CONCLUSION

The DCEA is a symmetric key encryption algorithm which uses the idea of generating sub keys using MD5 hash algorithm. This uses the idea of single point crossover genetic algorithm to generate 8 more sub keys from 4 sub keys. DCEA

outputs an average avalanche effect of 0.97 out of 1. In comparison to DCA, DCEA has higher throughput which results in a higher encryption speed of DCEA. The algorithm is also resistant to various attacks like frequency analysis attack, chosen plaintext attack, cipher text only attack and known plaintext attack which has been described in previous sections. Moreover the use of XOR logical operator inspire the implementation of the algorithm in both hardware and software mode.

REFERENCES

[1] Daniar Heri Kurniawan and Rinaldi Munir “ Double Chaining Algorithm - A Secure Symmetric-key Encryption Algorithm” IEEE conference on Advanced Informatics: Concepts, Theory And Application (ICAICTA), 16-19 Aug. 2016, George Town, Malaysia

[2] Tingyuan Nie, Chuanwang Song and Xulong Zhi, "Performance Evaluation of DES and Blowfish Algorithms", IEEE International Conference on Biomedical Engineering and Computer Science (ICBECS- 2010), pp. 1-4, 23-25 Apr 2010.

[3] Dr.Madhu Goel, Rupinder Kaur, "A Review of Some Popular Encryption Techniques", International Journal of Software and Web Sciences, 8(1), March-May 2014, pp. 41-45

[4] Ali Ahmad Milad, Hjh Zaiton Muda,Zul Azri Bin Muhamad Noh, Mustafa Almahdi Algaet, "Comparative Study of Performance in Cryptography Algorithms (Blowfish and Skipjack)", J. Computer Sci., 8 (7): 1191-1197, 2012

[5] Omari H. Ahmed, Al-Kasasbeh M. Basil, Al-Qutaish E. Rafa, Muhairat I. Muhammad, 2008, "A New Cryptographic Algorithm for the Real Time Applications ",Proceedings of the 7th WSEAS International Conference on INFORMATION SECURITY and PRIVACY (ISP '08)

[6] C. E. Shannon "Communication Theory of Secrecy Systems", Bell Systems Tech. Jr. Vol 28, pages 656-715, 1949

[7] Alan Kaminsky, Michael Kurdziel, Stanisław Radziszowski "An Overview of Cryptanalysis Research for the Advanced Encryption Standard" IEEE conference on MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010, 31 Oct.-3 Nov. 2010, San Jose, CA, USA

[8] Prerna Mahajan, Abhishek Sachdeva. "A Study of Encryption Algorithms AES, DES and RSA for Security". Global Journal of Computer Science and Technology Network, Web & Security, ISSN: 0975-4350, Volume 13, Issue 15, Version 1.0, March 2013. Global Journals Inc. (USA)

[9] Diaa Salama Abdul. Elminaam, Hatem Mohamed Abdul Kader and Mohie Mohamed Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, pp. 280-286, December 2008.

[10] Feistel, Horst (1973) "Cryptography and Computer Privacy". Scientific American 228

[11] Simar Preet Singh and Raman Maini "Comparison Of Data Encryption Algorithms", International Journal of Computer Science and Communication, Vol. 2, No. 1, January-June 2011, pp. 125-127

[12] M. Anand Kumar and Dr.S.Karthikeyan "Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms", I. J. Computer Network and Information Security, March 2012, pp. 22-28

[13] Sombir Singh, Sunil K Maakar and Dr. Sudesh Kumar "A Performance Analysis of DES and RSA Cryptography", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 3, May – June 2013

[14] Milind Mathur, Ayush Kesarwani "Comparison between DES, 3DES, RC2, RC6, Blowfish and AES", National Conference on New Horizons in IT - NCNHIT 2013, pp. 79-6

[15] K. F. Man, K. **S.** Tang, **S.** Kwong "Genetic Algorithms: Concepts and Applications", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 43, No. 5, OCTOBER 1996