Review Paper

Study and Literature Survey for use of Genetic Algorithm for Implementation of Advanced Encryption Standard

Amit Nevase[1], Prof. Nagnath Hulle[2]

[1]*Student, Department of Electronics & Telecommunication, GHRIET, Pune, Maharashtra, India*
[2]*Vice Principal, Department of Electronics & Telecommunication, GHRIET, Pune, Maharashtra, India*

Abstract*— The importance of cryptography applied to security in electronic data transactions has acquired an essential relevance over the past few years. Advanced Encryption Standard (AES) adopted by the National Institute of Standards and Technology (NIST) to replace existing Data Encryption Standard (DES), as the commonly used encryption algorithm in many security applications. To date, AES standard has different key size variants of 128, 192, and 256-bit. The longer bit keys provide more secure ciphered text output. AES generates keys through the properties of the Rijndael Algorithm instead of conventional method of the key generation. In this paper we study the literature of AES algorithm and key generation by using genetic algorithm and this generated key is used for formation of S-Box. This new approach for AES S-Box to enhance the complexity of the structure of S-Box, making AES stronger by using Dynamic S-Box.*

Keywords — Cryptography, Encryption-Decryption, AES, Genetic Algorithm, Genetic operators

I. INTRODUCTION

"Cryptography" refers to the change of information presentation from its unique structure into one another distinctive structure. The main goal is to make information more secured. The two main procedures are involved in Cryptography; the principal procedure is the encryption and decryption. Data Encryption Standard (DES) [12], the Elliptic Curve Cryptography (ECC) [12], the Advanced Encryption Standard (AES) [12] are the different cryptographic algorithm. The brute force were attempted to break such algorithms. Also some and side channel attacks are used to break such algorithms. The brute force attacks break the data encryption standard and made it uncertain algorithm. To replace Data Encryption Standard, National Institute of Standards and Technology (NIST) searched for another algorithm and Rijndael Algorithm is selected as next AES.

Nowadays, Advanced Encryption Standard was most secured cryptographic algorithm, used by NIST after Data Encryption Standard. The Advanced Encryption Standard algorithm has widely used in RFID cards, firewalls, routers, firmware, UART, ATM machines, cellular phones and big servers. The AES algorithm has vital importance and hence the main aim of project is to present new efficient and more secure hardware architecture implementation of AES algorithm. To implement hardware of AES algorithm, Field programmable Gate Arrays (FPGA) are used because they provide cryptographic algorithm agility, physical security, better performance. The main aim is high throughput which supports security and high bandwidth for various applications.

II. LITERATURE REVIEW

In [7], author makes use of genetic algorithms as a category of optimization algorithms. Genetic algorithms are used to solve issues through modeling with a simplified version of Genetic processes. There are numerous troubles for which a genetic set of rules method is relevant. It is, however, undetermined if cryptanalysis is this type of problems. Therefore, this work explores using genetic algorithm in cryptography. Each traditional cryptanalysis and GA based methods are implemented in software program. The outcomes are then compared the use of the metrics of elapsed time and percent of a success decryptions.

In [8], author uses genetic algorithms for the cryptanalysis of some of classical cryptosystems. Author provides the effects of applying some of GA implementations to diverse ciphers. The selection of appropriate mutation operators and fitness functions is also mentioned.

In [9], this paper offers new algorithms that simplify the introduction and growth technique of the encryption key of the AES algorithm, that's taken into consideration one of the maximum essential factors within the technique of encryption, via developing new key generator architectures that lets in us to have 10 specific keys. Those modifications, based totally on genetic set of rules and the linear feedback shift register algorithm, improve the performances and the efficiency of the encryption algorithm. The implementations of the Hardware and software program architectures are studied in Altera Cyclone II devices with the use of the VHDL language.

In [10], this paper, author proposes a genetic algorithm (GA) based symmetric key cryptosystem for encryption and decryption. right here, the text content and the user input (key) is converted into text matrix and key matrix

respectively. An additive matrix is generated via adding the text matrix and key matrix. A linear substitution function is applied at the additive matrix to produce the intermediate cipher. Then the GA features (crossover and mutation) are carried out on the intermediate cipher to produce the final cipher text. The proposed algorithm has simple steps, substitution followed by means of genetic crossover and mutation.

In [11], this paper gives application of genetic operators inside the field of cryptography. Author introduces a new encryption method in which the genetic operators crossover and mutation are used to encrypt messages, so one can guard the facts in the course of transmission. The goal is to provide a higher encryption method which is difficult for cryptanalysis.

In [12], author affords approach to locate the best fit element in the surroundings. As the author uses Genetic Algorithm to accomplish the project. The paper illustrates this innovative technique of key generation, also demonstrates its implementation. To attain even high standards of protection, Data encryption Standard has been used for verification and validation.

In [13], the importance of cryptography can be judged by means of the fact that it is used almost everywhere. It's far essential in e transactions. LAN data transfer, in databases and even at the same time as storing records in our very own pc. There are many strategies of cryptography. Author makes use of the deciding on key for the public key cryptography is an expansion procedure wherein various keys may be classified on the idea in their fitness function, makes genetic algorithms an amazing contender for the generation of the public key.

In [14], this paper author proposes a new approach of genetic algorithms (GA) with pseudorandom sequence to encrypt data stream. The feature of such an approach includes greater protection and high feasibility for easy integration with industrial multimedia transmission programs. The experimental effects of the proposed technique affirm that high throughput rate needed for real time protection.

In [15], this paper, author proposes a technique based totally on genetic algorithms (GA) that's used to provide a new encryption approach through exploitation the powerful features of the crossover and mutation operations of GA. Many issues can be solved the usage of genetic algorithms through modeling a simplified version of genetic techniques.

III. ADVANCED ENCRYPTION STANDARD

The two Scientists from Belgian Vincent Rijmen and John Daemen [1] was publish a cipher Rijndael in 1998. After the failure of Data Encryption Standard, the Rijndael [1] was selected as Advanced Encryption Standard. The AES overcomes the disadvantages of DES. Under the Federal Information Processing Standards Publication 197 (FIPS), the National Institute of Standards and Technology (NIST), discuses all the information of AES. The Advanced Encryption Standard has a data size 128 bits i.e. 16 bytes with different key sizes of 128 bits, 192 bits and 256 bits. The key size is depends upon the number of encryption rounds. The 128 bit key size for 10 round, 192 for 12 and 256 for 14 rounds. The 128 bit data is formed in 4 X 4 array, which is called State Array, is used in encryption process. The initial data is converted into state array during the encryption process, after each round these data is changing until reached to final cipher text. During the decryption process, these cipher text array is keep changing to obtain original data. In every round of encryption and decryption process, new round key is generated using the initial key applied. The round structure of Advanced Encryption Standard shown in fig. 1

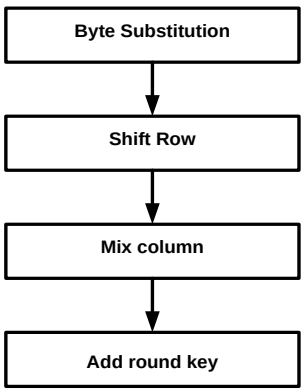

Figure 1 AES Round Structure

The Advanced Encryption Standard structure with round 0 through 10 is shown in Figure 2.

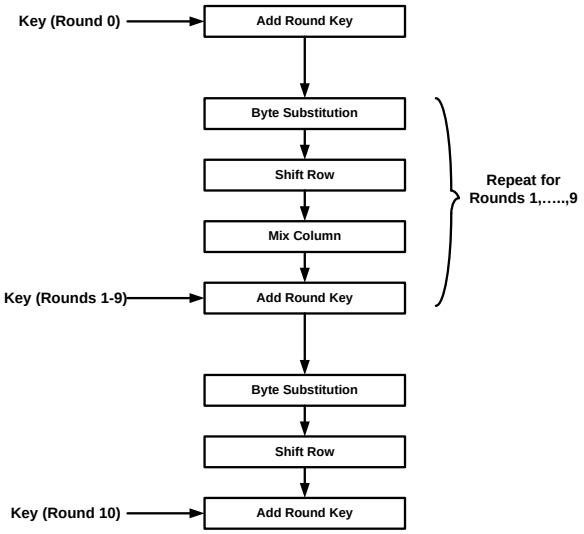

Figure 2 AES Structure

Each block consists of 128 bits, and these are divided into 16, 8 bit bytes. Each of the operations acts upon these 8 bit bytes in a 4×4 matrix:

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Note that each (a_i, j) is an 8-bit byte, viewed as elements of GF (2^8). The arithmetic operators take advantage of the Galois Field rules defined in the Rijndael algorithm, an example is that of addition that is implemented by XOR. The model of the finite field GF(2^8) depends on the choice of an irreducible polynomial of degree 8, which for Rijndael is:

$$X^8 + X^4 + X^3 + 1$$

Each of the round operations requires a specific mathematical exploration.

a) Byte Substitution

Byte Substitution [1] requires that for each input data block *a*(3,3), we look up a table of substitutions and replace the bytes to produce a new matrix *b*(3,3). The way it works, is that for each input byte abcdefgh, we look up row abcd and column efgh and use the byte at that location in the output.

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \Rightarrow \text{Byte Substitution}$$

$$\Rightarrow \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

The complete byte substitution table is defined using the following Table 1:

Table 1: AES Byte Substitution Table

099	124	119	123	242	107	111	197	048	001	103	043	254	215	171	118
202	130	201	125	250	089	071	240	173	212	162	175	156	164	114	192
183	253	147	038	054	063	247	204	052	165	229	241	113	216	049	021
004	199	035	195	024	150	005	154	007	018	128	226	235	039	178	117
009	131	044	026	027	110	090	160	082	059	214	179	041	227	047	132
083	209	000	237	032	252	177	091	106	203	190	057	074	076	088	207
208	239	170	251	067	077	051	133	069	249	002	127	080	060	159	168
081	163	064	143	146	157	056	245	188	182	218	033	016	255	243	210
205	012	019	236	095	151	068	023	196	167	126	061	100	093	025	115
096	129	079	220	034	042	144	136	070	238	184	020	222	094	011	219
224	050	058	010	073	006	036	092	194	211	172	098	145	149	228	121
231	200	055	109	141	213	078	169	108	086	244	234	101	122	174	008
186	120	037	046	028	166	180	198	232	221	116	031	075	189	139	138
112	062	181	102	072	003	246	014	097	053	087	185	134	193	029	158
225	248	152	017	105	217	142	148	155	030	135	233	206	085	040	223
140	161	137	013	191	230	066	104	065	153	045	015	176	084	187	022

For Example: If the input data byte was 8A, then this in binary terms is

1000 1010

So the row required is 8 (1000) and the column required is A (1010). From this we can read off the resulting number from the table:

126=0111 1110= 7E

This is illustrated in the figure 4.

We can see that this is a bit shuffling operation that is simply moving bytes around in a publicly defined manner that does not have anything to do with a key. Also the individual bits within the byte are not changed per se. this is a bytewise operation.

	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
0	099	124	119	123	242	107	111	197	048	001	103	043	254	215	171	118
1	202	130	201	125	250	089	071	240	173	212	162	175	156	164	114	192
2	183	253	147	038	054	063	247	204	052	165	229	241	113	216	049	021
3	004	199	035	195	024	150	005	154	007	018	128	226	235	039	178	117
4	009	131	044	026	027	110	090	160	082	059	214	179	041	227	047	132
5	083	209	000	237	032	252	177	091	106	203	190	057	074	076	088	207
6	208	239	170	251	067	077	051	133	069	249	002	127	080	060	159	168
7	081	163	064	143	146	157	056	245	188	182	218	033	016	255	243	210
8	205	012	019	236	095	151	068	023	196	167	126	061	100	093	025	115
9	096	129	079	220	034	042	144	136	070	238	184	020	222	094	011	219
A	224	050	058	010	073	006	036	092	194	211	172	098	145	149	228	121
B	231	200	055	109	141	213	078	169	108	086	244	234	101	122	174	008
C	186	120	037	046	028	166	180	198	232	221	116	031	075	189	139	138
D	112	062	181	102	072	003	246	014	097	053	087	185	134	193	029	158
E	225	248	152	017	105	217	142	148	155	030	135	233	206	085	040	223
F	140	161	137	013	191	230	066	104	065	153	045	015	176	084	187	022

Figure 4 Byte Substitution Example

b) **Shift Row**

The shift rows [1] step is a byte transposition that cyclically left shifts the rows of the state over different offsets. Row 0 is shifted with offset 0, row 1 is shifted with offset 1, row 2 is shifted with offset 2 and row 3 is shifted with offset 3.

$$\begin{pmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix} = \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,1} & b_{1,2} & b_{1,3} & b_{1,0} \\ b_{2,2} & b_{2,3} & b_{2,0} & b_{2,1} \\ b_{3,3} & b_{3,0} & b_{3,1} & b_{3,2} \end{pmatrix}$$

c) Mix Column

The Mix column [1] step is a bricklayer permutation operating on the state column by column. In other words the mix columns function is a series of specific multiplications:

$$\begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{pmatrix} = \begin{pmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{pmatrix} * \begin{pmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix}$$

Where:
‘01’ = 00000001
‘02’ = 00000010
‘03’ = 00000011

All multiplications are GF(2^8) and this transformation is invertible.

d) Add Round Key

In the AddRoundKey [1], a Round Key is added to the State, result in the operation of the MixColumns transformation - by a simple bitwise XOR operation. The RoundKey of each round is generated from the main key using the Key Expansion algorithm. The encryption and decryption algorithm needs eleven 128-bit RoundKey. The final operation in each round is to add the key using the following function:

$$\begin{pmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3} \end{pmatrix} = \begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{pmatrix} \oplus \begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix}$$

IV. DYNAMIC S BOX

Static S-box means the same S-box will be used in each round while the key-dependent S-box means the S-box changes in each round depending on the key and different number of rounds. The attackers may study Fixed S-box and find weak points. But in case of key-dependent S-Box approach, the attackers cannot do any offline analysis of one particular set of S-box.

A new AES-like design for key- dependent AES using S-box rotation. The algorithm involves key expansion algorithm (Genetic Algorithm) together with S-box rotation and this property can be used to make the S-box key-dependent (Dynamic S-Box)[18]. Therefore it provides a better security to the block cipher.

V. GENETIC ALGORITHM

In 1859 Charles Darwin (1809-82) [11] published an extremely controversial book of title is “On the origin of species by means of natural selection, or the preservation of favored races in the conflict of life”, which is well known as The origin of species. At the same time, Gregor Mendel (1822-84) [11] searched the inheritance of characteristics, or traits, in his experiments with pea plants. By means of examination of hybrids from different strains of plant he obtained some notion of the interactions of characters. In spite of the fact that Mendel's investigations established the frameworks for the investigation of genetics, it was not until 30 years after his demise that Walter Sutton (1877-1916) [11] found that genes were a part of chromosomes in the nucleus. However, Darwin's theory emphasized the role of continuous variation within species. Conversely, particular contrasts between species are not uncommon in nature, i.e. discontinuous variation. Hugo de Varis (1848-1935) [11] observed that in a population of developed plants, strikingly different variations would periodically show up. To clarify this discontinuous variation, de Varis built up a theory of mutation. Superficially, the new investigation of genetics appeared to support the mutation theory of advancement against conventional Darwinism.
The Genetic algorithms are random search and optimization methods based upon the analogy of natural biological origin. Genetic algorithms are the part of the evolutionary algorithms which additionally consists of evolutionary programming, evolution techniques and genetic programming. evolutionary algorithms operates with a population of ability solutions to a problem. It makes use of principle of survival of best fit solution. Mutation and reproduction to produce solution. At every new iteration of an Evolutionary Algorithm, a new generation of alternative solutions are created by the processes of selection and reproduction

a) Elements of Genetic Algorithm

The Goldberg evaluates simple genetic algorithm (SGA). It has three basic elements selection, crossover and mutation and is to illustrate in flow chart shown in figure 5.

1) Population Generation

The procedure of Genetic Algorithms, for the most part, begins with a populace which is arbitrarily created and is made out of a few chromosomes.

2) Selection

Selection is the process of finding the particular individual or number of offspring, alternative solutions for reproduction.

3) Crossover

After the generation of population, the important parameter of genetic algorithm is to connect generated population. Likewise there are distinctive sorts of crossover, for example, one point crossover or two point crossover or uniform crossover. Crossover operator is used to generate best fit generation than previous one.

4) Mutation

The mutation is arbitrary process where one factor of a gene is replaced by different to generate a new genetic structure.

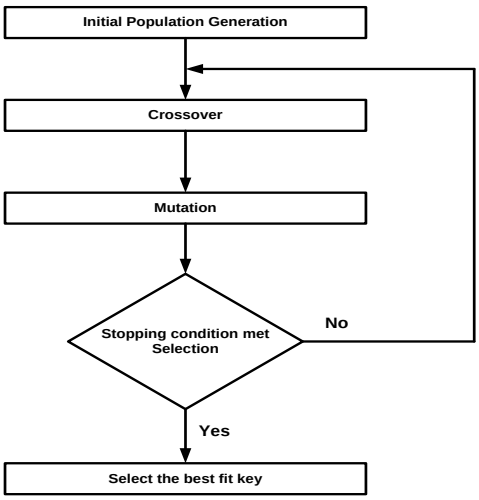

Figure 4 Elements in GA

VI. CONCLUSIONS

In this paper we present the study of Advance Encryption Standard algorithm also study the process of byte substitution, shift row transformation, mix column step and add round key transformation. Also we present the study of literature survey of Genetic algorithm for key generation. The key generation will be takes place by major elements of Genetic Algorithms like crossover, mutation and selection. The generated key is used for the making the byte substitution key dependent and these newly generated Dynamic S-Boxes are used in the different rounds in Advance Encryption Standard (AES) process.

REFERENCES

[1] Joan Daemen and Vincent Rijimen. *The Design of Rijindael.* Springer,2002, pp-31-50.

[2] X. Zhang and K. K. Parhi, "Implementation approaches for the advanced encryption standard algorithm", *IEEE Circuits Syst. Mag.*, vol. 2, no. 4, pp.24 -46 2002.

[3] Alireza Hodjat, Ingrid Verbauwhede. "Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processors". *IEEE TRANSACTIONS ON COMPUTERS*, VOL. 55, NO. 4, 366-372, APRIL 2006.

[4] Pawel Chodowiec, Po Khuon, Kris Gaj. "Fast implementations of secret-key block ciphers using mixed inner- and outer-round pipelining". *International symposium on Field programmable gate arrays, Monterey, CA*, February 2001.

[5] K. Gaj and P. Chodowiec, "Comparison of the hardware performance of the AES candidates using reconfigurable hardware", *Proc. 3rd AES Conf. (AES3)*, 2000 [online] Available: item>http://csrc.nist.gov /encryption/aes/round2/conf3/aes3papers.html</item>

[6] Bin Liu, Bevan M. Baas. "Parallel AES Encryption Engines for Many-Core Processor Arrays". IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 3, 536-547, MARCH 2013.

[7] Bethany Delman, Genetic Algorithms in Cryptography, MS Thesis 2004.

[8] Karel P.Bergmann,Renate Scheidler, Christian Jacob. "Cryptanalysis using Genetic Algorithms", GECCO 2008.

[9] Sliman Arrag, Abdellatif Hamdoun, Abderrahim Tragha. "Replace AES Key Expansion Algorithm By Modified Genetic Algorithm". *Applied Mathematical Sciences*, Vol. 7, No. 144, 7161-7171, 2013.

[10] Sindhuja K , Pramela Devi S. "A Symmetric Key Encryption Technique Using Genetic Algorithm". *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 5 (1), 414-416, 2014.

[11] Amritha Thekkumbadan Veetil. “An Encryption Technique Using Genetic Operators”. *International Journal Of Scientific & Technology Research*, Vol. 4, Issue 07, 202-203, July 2015.

[12] Sania Jawaid, Adeeba Jamal. “Generating the Best Fit Key in Cryptography using Genetic Algorithm”. *International Journal of Computer Applications, Volume 98 – No.20 33-39, July 2014*

[13] Sonia Goyat. “Genetic Key Generation For Public Key Cryptography”. *International Journal of Soft Computing and Engineering (IJSCE)*, Volume-2, Issue-3, 231-233, July 2012.

[14] Anil Kumar and M. K. Ghose, Overview of Information Security Using Genetic Algorithm and Chaos, *Information Security Journal: A Global Perspective*, 18:306–315, 2009.

[15] Ankita Agarwal. “Secret Key Encryption Algorithm Using Genetic Algorithm”. *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 2, Issue 4, 216-218, April 2012.

[16] A.M.S.Zalzala and P.J.Fleming. *Genetic Algorithms in Engineering Systems.* IEE Control Engineering Series 55,1997, pp-1-41.

[17] Serge Vaudenay. *A Classical Introduction to Cryptography-Applications for Communications Security*. Springer,2006, pp-1-60.

[18] Ashwak alabaichi and Adnan Ibrahem Salih. “Enhance Security of Advance Encryption Standard Algorithm Based on Key Dependent S-Box”, Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference. Sierre,2015, pp-44-53.