Encryption and Decryption Using Automata Theory

Zubair Saqib[1], Murtza Ahmad Shahid[2] and Muhammad Umair Ashraf[3]
[1,2,3]University of Engineering and Technology, Lahore, Pakistan

***Abstract*– Automata theory is useful tool used in artificial intelligence and humanoid robots design. It can also be used in the encryption and decryption of communication going on between two locations. It's very useful tool Turing machine is first designed to decrypt the communication going on between German warheads in World War II and nowadays every sensitive information is sent in encrypted form. This paper represents the design of the encryption and decryption using Turing machine.**

***Keywords*– Automata Theory, Encryption, Decryption and Turing Machine**

I. INTRODUCTION

In the theoretical computer science and mathematics theory of computation is the branch that deals with if and how proficiently complications can be solved on a model of computation, using a routine. The field is alienated into three key outlets: automata theory, computability theory and computational complexity theory.

Automata theory is the study of intellectual mechanisms (or more fittingly, nonfigurative 'mathematical' machines or systems) and the computational complications that can be explained using these machines. These nonconcrete machines are called automata. Automata comes from the Greek word (Αυτόματα) which means that something is exploit something by itself. Automata theory is also narrowly linked to formal language theory, as the automata are frequently categorized by the class of formal languages they are able to distinguish. An automaton can be a restricted demonstration of a formal language that may be an unbounded set.

In order to achieve an arduous study of reckoning, computer scientist's work with a mathematical generalization of computers called a model of computation. There are numerous replicas in use, but the most regularly studied is the Turing machine. Computer scientists study the Turing machine because it is simple to frame, can be evaluated and used to prove consequences, and since it exemplifies what many deliberate the supreme commanding conceivable "judicious" model of computation. It might seem that the theoretically unlimited reminiscence ability is an unrealizable feature, but any decidable problem answered by a Turing machine will every time require only a limited volume of memory. So in opinion, any problem that can be explained (Absolute) by a Turing machine can be cracked by a computer that has a limited extent of reminiscence. The encryption and decryption is not an exception. In this paper we are doing it using computer theory tools. First we encrypt the communication string using our encryptor formed by using Turing machine in 3-state busy beaver form and then we decrypt the string back to its original form using decryptor that we designed. Experimental run of the Turing machine is carried out and shown for the encryption and decryption in this paper.

II. EXISTING ENCRYPTION AND DECRYPTION WORK

There are many encryption algorithms available now a days and each of them have its own encryption scheme one of which is the ASCII base encryption of the string in which the data is encrypted by using ASCII values of the data to be encrypted but this algorithm have its own restriction that the algorithm only operates when the length of input string and the length of the key are the same.

Another algorithm that is used to improve the data security is based on the block cipher concept. In this algorithm different logic operations are used such as XOR and different shift operations .This algorithm is also very efficient and secured and is often used.

Some of them are very common in attaining data security at a great degree like AES and Blowfish. But, as security level is enlarged, the time and complexity of algorithm is also increased. This is the key cause of decreasing the swiftness and effectiveness of the encryption system. A new encryption algorithm that is "Byte – Rotation Encryption Algorithm (BREA)" with "Parallel Encryption Model" which boosts the security as well as speed of the encryption scheme. The BREA is applied on different chunks of plaintext and effects in parallel manner over multithreading concept of single processor system.

Nowadays, a paper encryption technology is used which encrypts the data written on the paper by using techniques like distortions, tilting, expansion and shrinkage this change the data written on the paper so that the normal human eye cannot see, or more precisely we may say, understand the data unless proper decryption is performed on the paper or we use some kind of medium that uses the encryption scheme like the camera of the cell phone can decrypt the encryption on the paper and show the human understandable image or data on the screen. This is only one method there are many other methods in paper encryption for encryption and decryption of the information present on the paper.

Many operating systems have now-a-days already include the encryptions in them like Apple includes a volume encryption software named FileVault 2 in their operating system. While the previous Version of FileVault (presented

with Mac OS X 10.3) only encrypted the home folder, FileVault 2 can encrypt the full capacity containing the operating system (this is commonly referred to as full disk encryption).

Another encryption scheme for users who wants security level and processing level is a block cipher which is a derivation on the fiestle network Architecture. The algorithm provides security levels and their parallel processing levels by using several keys for the encryption/decryption method. This capability is accomplished by using fuzzy logic. The results of the proposed encryption algorithm will be examined by paralleling with other surviving encryption Algorithms.

There is an available application which even encrypts image files, data files and documentation files. This application uses simple key generation method of arbitrary digit cohort and grouping. The final Encryption is a binary one accomplished through turning of bits and XOR operation applied on each chunk of data in any file using a symmetric decimal key. The key Generation and Encryption are all done by the system itself after clicking the encryption button with clearness to the user. The same encryption key is also used to decrypt the encrypted binary file.

III. ENCRYPTION AND DECRYPTION

First we use a simple encryption scheme in which we encrypt our string using static form of encryption in which a word is simply replaced by its encrypted replica and the output is shown in Table 1. Here is the design of the static encryptor that encrypt the string using simple encryption scheme.

Table 1: Encryption and decryption

Symbol	Replica	Symbol	Replica
A	00001	N	01110
B	00010	O	01111
C	00011	P	10000
D	00100	Q	10001
E	00101	R	10010
F	00110	S	10011
G	00111	T	10100
H	01000	U	10101
I	01001	V	10110
J	01010	W	10111
K	01011	X	11000
L	01100	Y	11001
M	01101	Z	11010
		Δ	Δ

Post-Turing instruction-abbreviations used for this example:

R= Move tape right. L= Move tape Left.

Z= Read tape symbol.

H=Stop machine after successful encryption.

S= Replace output of instruction ‘F’ with original symbol on Tape.

F= Replace tape symbol with its binary form replica.

Jxxx=unconditionally go to the instruction xxx.

JΔxxx= If tape symbol is ‘Δ‘go to ‘M’ state else go to next state.

C= Check for respected replica of the symbol from replica set.

JTxxx= Go to next instruction if the respected replica is present in the instruction ‘C’.

E= Stop the machine giving an error.

M= Read 10 tape symbols if all are ‘Δ’ go to ‘H’ else go to next instruction.

Sample input:

This year our profit is six million

Sample output:

10100010000100110011Δ11001001010000110010Δ01111 1010110010Δ10000100100111100110010Δ01101000100110 011Δ100110100111000Δ01101010010110001100010010111 101110 ΔΔΔΔΔΔΔΔΔΔ

Now we send this encrypted string at our destination and then we decrypt it back to its original form using the static decryptor that uses a simple decryption scheme.

Post-Turing instruction-abbreviations used for this example:

R= Move tape right. L= Move tape Left.

Z= Read tape symbol.

H= Stop machine after successful encryption.

E= Stop the machine giving an error.

C= Check for respected symbol for the replica.

F= Place symbol at the start of the tape if another symbol is already present place it next to the available place.

P= Read 5 successive 0’s and 1’s combination and send it to the next state.

Jxxx = unconditionally go to the instruction Jxxx

JΔxxx= If tape symbol is ‘Δ‘go to ‘M’ state else go to next state.

JT= Go to next instruction if the respected replica is present.

M= Read 10 tape symbols if all are ‘Δ’ go to ‘H’ else go to next instruction.

Sample input:

10100010000100110011Δ11001001010000110010Δ01111 1010110010Δ10000100100111100110010Δ01101000100110 011Δ100110100111000Δ01101010010110001100010010111 101110ΔΔΔΔΔΔΔΔΔΔ

Sample output:

This year our profit is six million

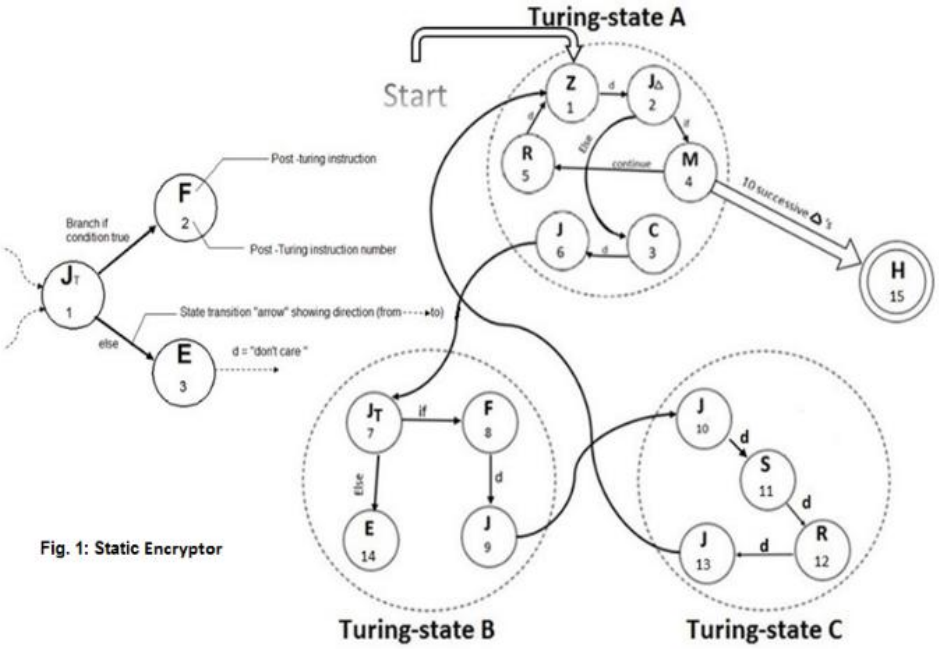

Fig. 1: Static Encryptor

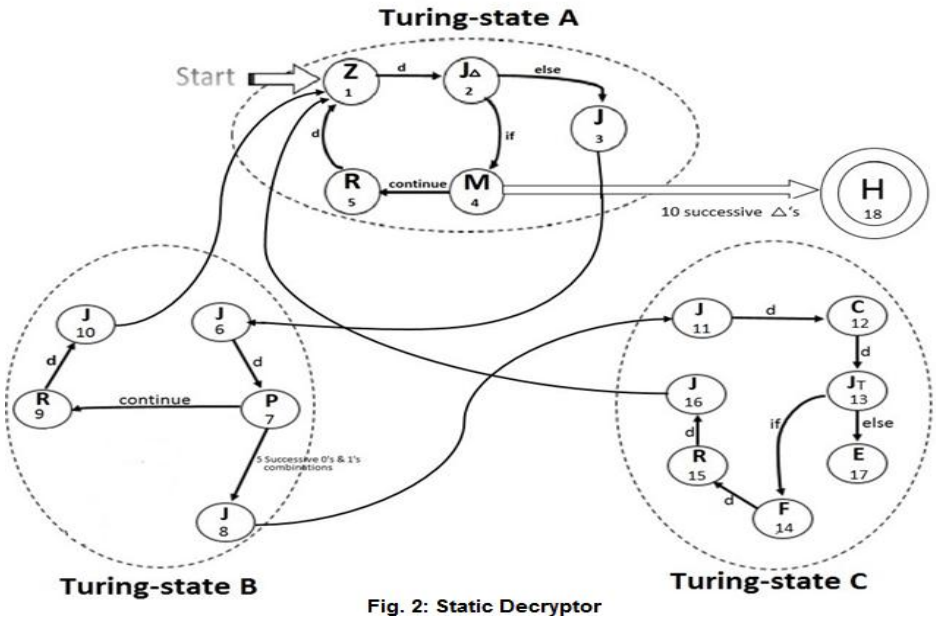

Fig. 2: Static Decryptor

This will decrypt our string back to its original form and we have successfully done a secure communication between source and destination.

Now we move onto some more difficult encryption scheme. Suppose we are designing an encryption scheme for the army than the information that is sent is very sensitive and its lost cannot be bare so we use a dynamic encryption scheme in which we replace the symbol with its replica but we also insert a small amount of garbage in it so that it is not easy to recognize the original message.

Garbage Set:

{ Ò Ø ü â ä à å ç £ ø º ª ¿ ® ¬ ¡ © ¦ ¢ ¤ ð ß! @ # $ % ^ & * ( ) _ - + = { } : " ? > < , . / ';] [\ | **}**

Assume a case in which the two army battalions are trying to communicate with each other using their walkie-talkie and they are communicating with each other. We have designed the walkie-talkie such as it gets the message in the form of voice and place it on the input tape of the decryptor machine that we have embed in the walkie-talkie which encrypts the message and then send it in the form of signals in the medium i.e. air using the transmitter at certain frequency.

Post-Turing instruction-abbreviations used for this example:

R= Move tape right. L= Move tape Left.

Z= Read tape symbol.

H= Stop machine after successful encryption.

S= Replace output of instruction 'P' with original symbol on tape.

P= Concatenate 5 random characters from garbage set at the end of replica.

F= Replace tape symbol with its binary form replica.

Jxxx= unconditionally go to the instruction xxx.

JΔxxx= If tape symbol is 'Δ'go to 'M' state else go to next state.

C= Check for respected replica of the symbol from replica set.

JTxxx= Go to next instruction if the respected replica is present in the instruction 'C'.

E= Stop the machine giving an error.

M= Read 10 tape symbols if all are 'Δ'go to 'H' else go to next instruction.

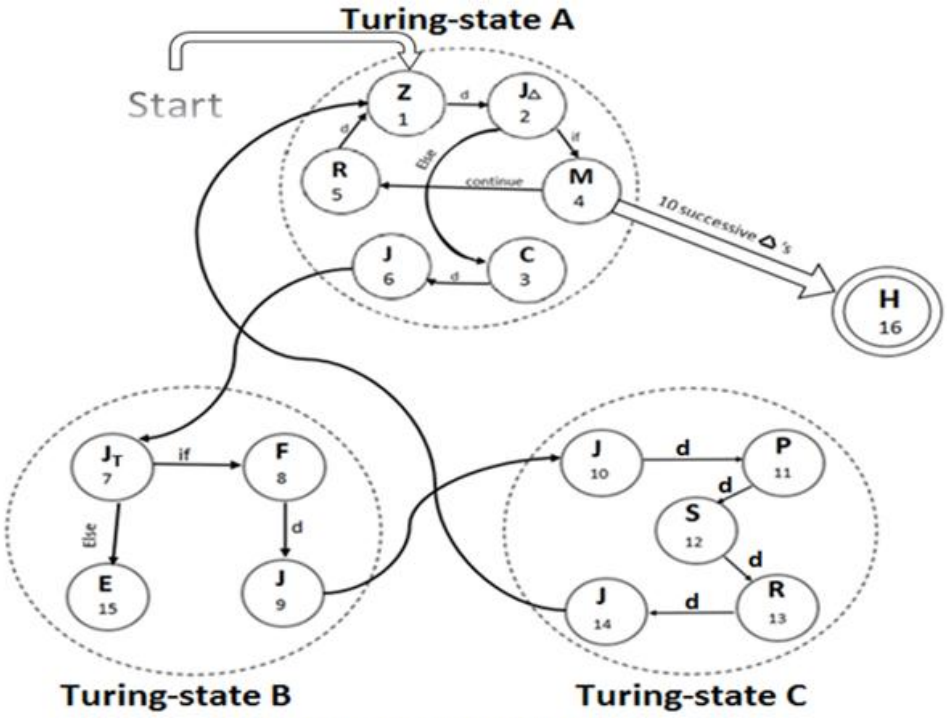

Fig. 3: Dynamic Encryptor(Less garbage)

Sample input:

Vanguard six this is bravo one we are ready

Sample output:

10110@*^$(00001>"{^$01110&%+<#00111,.;/]00001}:">*10101@$#&^10010.[;',00100Δ10011$^#*%01001!)(&^11000Δ10100!)*%*01000!^%&%01001<:{(&10011Δ01001)*^$@10011Δ00010<>)(^10010*)^$*00001*&*%$10110_)(&*01111Δ01111<>?:"01110$%^#@00101Δ10111*(&%*00101Δ00001;'[].10010$%#^%00101Δ10010@#$#@00101(^*%#00001;'[].00100$&#*%11001(^*%#00001;'[]!ΔΔΔΔΔΔΔΔΔΔ

At the destination the signals are received via receiver and it converts the signal back to message which is place on the input tape of our embedded decryptor of the walkie-talkie that converts it back to its original form and then a voice message can be heard from the speaker of the walkie-talkie.

Post-Turing instruction-abbreviations used for this example:

R= Move tape right. L= Move tape Left.
Z= Read tape symbol.
H= Stop machine after successful encryption.
E= Stop the machine giving an error.
C= Check for respected symbol for the replica.
F= Place symbol at the start of the tape if another symbol is already present place it next to the available place.
P= Read 5 successive 0's and 1's combination and send it to the next state.
Jxxx= unconditionally go to the instruction Jxxx
JΔxxx= If tape symbol is 'Δ'go to 'M' state else go to next state.
JT= Go to next instruction if the respected replica is present.
M= Read 10 tape symbols if all are 'Δ 'go to 'H' else go to next instruction.
G= any character other than 0's and 1's and 'Δ' discard them.

Sample input:

10110@*^$(00001>"{^$01110&%+<#00111,.;/]00001}:">*10101@$#&^10010.[;',00100Δ10011$^#*%01001!)(&^11000Δ10100!)*%*01000!^%&%01001<:{(&10011Δ01001)*^$@10011Δ00010<>)(^10010*)^$*00001*&*%$10110_)(&*01111Δ01111<>?:"01110$%^#@00101Δ10111*(&%*00101Δ00001;'[].10010$%#^%00101Δ10010@#$#@00101(^*%#00001;'[].00100$&#*%11001(^*%#00001;'[]!ΔΔΔΔΔΔΔΔΔΔ

Sample output:

Vanguard six this is bravo one we are ready

Now suppose another case in which the communication is going on between the two army base via booster which means we have large traffic of communication taken place through them which is also very sensible and cannot be compromised at any cost so we have to make our encryption scheme more strong and for this we are increasing the amount of garbage that insert in the string and due to which our encrypted form become very large and very difficult to get the original message because even the small message of 10 to 20 words can become a message of 100 to 120 words but we don't have to worry about the size of the message because we are communicating via towers which means we have lots and lots of bandwidth.

So the two army bases are trying to communicate with each other using their communication media. We have designed the media such as it gets the message in the form of voice via micro phone and place it on the input tape of the decryptor machine that we have embed in the communication media which encrypts the message and then send it in the form of signals via transmitting tower in the medium i.e. air at the certain frequency.

Post-Turing instruction-abbreviations used for this example:

R = Move tape right. L = Move tape Left.
Z = Read tape symbol.
H = Stop machine after successful encryption.
S = Replace output of instruction 'P' with original symbol on tape.
P = Concatenate 20 random characters from garbage set at the end of replica.
F = Replace tape symbol with its binary form replica.
Jxxx = unconditionally go to the instruction xxx.
JΔxxx = If tape symbol is 'Δ'go to 'M' state else go to next state.
C = Check for respected replica of the symbol from replica set.
JTxxx= Go to next instruction if the respected replica is present in the instruction 'C'.
E= Stop the machine giving an error.
M= Read 10 tape symbols if all are 'Δ' go to 'H' else go to next instruction.

Sample input:

Tombstone three in bound from the north coming in high requesting the cover fire on my retrieve

Sample output:

10100$%#^%&*^%$(*&^)#$%#@01111<>:"{)(*()^&^%$%*&^$01101>?:"}%^$%#)(*&)%&#@!00010,.;]'<>:"{&(^*^#%$@%10011*&^(&$%#$%(*&^%#$%&(10100#$^%#)(^(^@>{($:"}>?01111$*&^()&%#%<>*)^@#%$#01110!@#$%^&*()(*&^%$%^&*00101Δ10100>?:{})(*(&%^&%$#$*&^01000(*&(^&%$^$#%*&^$%^*%10010#$^%#)(^(^@>{($:"}>?00101>?:"}%^$%#)(*&)%&#@!00101Δ01001,.;]'<>:"{&(^*^#%$@%01110Δ00010*&^(&$%#$%(*&^%#$%&(01111*&^(&$%#$%(*&^%#$%&(10101#$^%#)(^(^@>{($:"}>?01110$%#^%&*^%$(*&^)#$%#@00100Δ00110$%#^%&*^%$(*&^)#$%#@10010*&^(&$%#$%(*&^%#$%&(011

11*&^(&$%#$%(*&^%#$%&(01101Δ10100$%#^%&*^%$(*&^)#$%#@01000>?:"}%^$%#)(*&)%&#@!00101Δ01110$%#^%&*^%$(*&^)#$%#@01111,.;]'<>:"{&(^*^#%$@%10010>?:"}%^$%#)(*&)%&#@!10100*&^(&$%#$%(*&^%#$%&(01000Δ00011>?:"}%^$%#)(*&)%&#@!01111*&^(&$%#$%(*&^%#$%&(01101>?:"}%^$%#)(*&)%&#@!01001,.;]'<>:"{&(^*^#%$@%01110>?:"}%^$%#)(*&)%&#@!00111Δ01001*&^(&$%#$%(*&^%#$%&(01110Δ01000>?:"}%^$%#)(*&)%&#@!01001*&^(&$%#$%(*&^%#$%&(00111#$^%#)(^(^@>{($:"}>?01000Δ10010>?:"}%^$%#)(*&)%&#@!00101>?:"}%^$%#)(*&)%&#@!10001!@#$%^&*()(*&^%$%^&*10101#$^%#)(^(^@>{($:"}>?001001>?:"}%^$%#)(*&)%&#@!10011>?:"}%^$%#)(*&)%&#@!10100$%#^%&*^%$(*&^)#$%#@01001#$^%#)(^(^@>{($:"}>?01110>?:"}%^$%#)(*&)%&#@!00111Δ10100$%#^%&*^%$(*&^)#$%#@01000>?:"}%^$%#)(*&)%&#@!00101Δ00011$%#^%&*^%$(*&^)#$%#@01111>?:"}%^$%#)(*&)%&#@!10110*&^(&$%#$%(*&^%#$%&(00101,.;]'<>:"{&(^*^#%$@%10010Δ00110,.;]'<>:"{&(^*^#%$@%01001!@#$%^&*()(*&^%$%^&*10010>?:"}%^$%#)(*&)%&#@!00101Δ01111,.;]'<>:"{&(^*^#%$@%01110Δ01101>?:"}%^$%#)(*&)%&#@!11001Δ10010,.;]'<>:"{&(^*^#%$@%00101$%#^%&*^%$(*&^)#$%#@10100!@#$%^&*()(*&^%$%^&*10010!@#$%^&*()(*&^%$%^&*01001>?:"}%^$%#)(*&)%&#@!10110#$^%#)(^(^@>{($:"}>?00101>?:"}%^$%#)(*&)%&#@!ΔΔΔΔΔΔΔΔΔΔΔΔ

At the destination the signals are received via receiving tower and it converts the signal back to message which is place on the input tape of our embedded decryptor of the communication media that converts it back to its original form and then a voice message can be heard from the speaker of the communication media.

Post-Turing instruction-abbreviations used for this example:

R= Move tape right.
L= Move tape Left.
Z= Read tape symbol.
H= Stop machine after successful encryption.
E= Stop the machine giving an error.
F= Place symbol at the start of the tape if another symbol is already present place it next to the available place.
C= Check for respected symbol for the replica.
P= Read 5 successive 0's and 1's combination and send it to the next state.
Jxxx= unconditionally go to the instruction Jxxx
JΔxxx= If tape symbol is 'Δ'go to 'M' state else go to next state.
JT= Go to next instruction if the respected replica is present.
M= Read 10 tape symbols if all are 'Δ' go to 'H' else go to next instruction.
G= any character other than 0's and 1's and 'Δ' discard them.

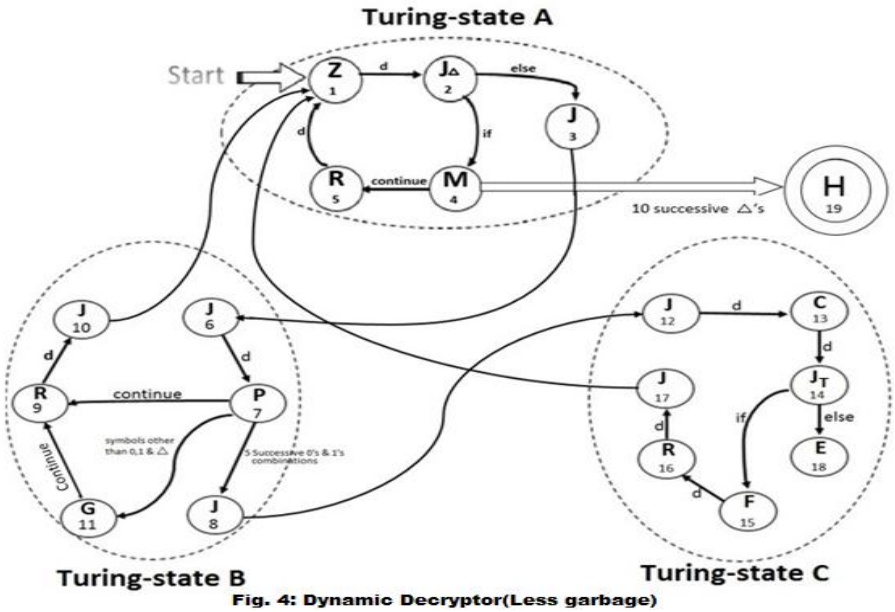

Fig. 4: Dynamic Decryptor(Less garbage)

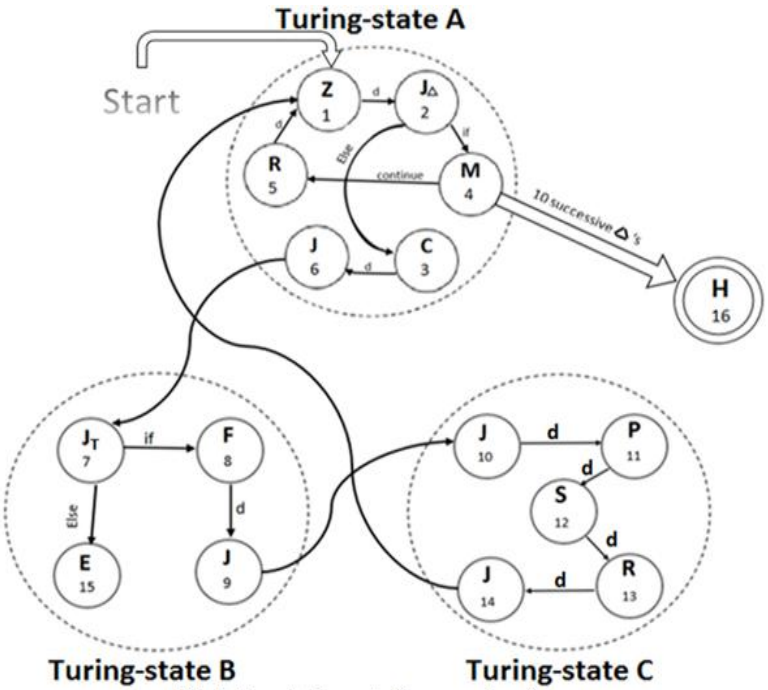

Fig. 5: Dynamic Encryptor(Excess garbage)

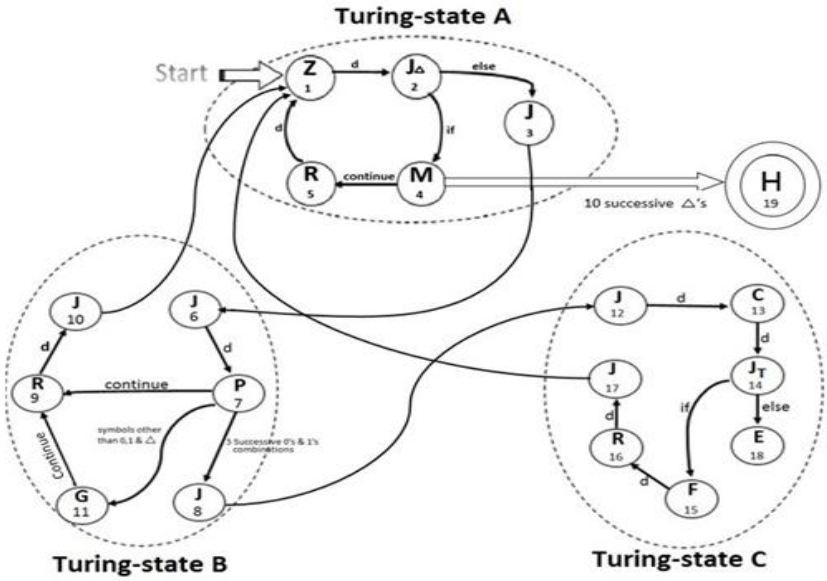

Fig. 6: Dynamic Decryptor(Excess garbage)

Sample input:

10100$%#^%&*^%$(*&^)#$%#@01111<>:"{)(*()^&^%$%*&^$01101>?:"}%^$%#)(*&)%&#@!00010,.;]'<>:"{&(^*^#%$@%10011*&^(&$%#$%(*&^%#$%&(10100#$^%#)(^(^@>{($:"}>?01111$*&^()&%#%<>*)^@#%$#01110!@#$%^&*()(*&^%$%^&*00101Δ10100>?:{})(*(&%^&%$#$*&^01000(*&(^&%$^$#%*&^$%^*%10010#$^%#)(^(^@>{($:"}>?00101>?:"}%^$%#)(*&)%&#@!00101Δ01001,.;]'<>:"{&(^*^#%$@%01110Δ00010*&^(&$%#$%(*&^%#$%&(01111*&^(&$%#$%(*&^%#$%&(10101#$^%#)(^(^@>{($:"}>?01110$%#^%&*^%$(*&^)#$%#@00100Δ00110$%#^%&*^%$(*&^)#$%#@10010*&^(&$%#$%(*&^%#$%&(01111*&^(&$%#$%(*&^%#$%&(01101Δ10100$%#^%&*^%$(*&^)#$%#@01000>?:"}%^$%#)(*&)%&#@!00101Δ01110$%#^%&*^%$(*&^)#$%#@01111,.;]'<>:"{&(^*^#%$@%10010>?:"}%^$%#)(*&)%&#@!10100*&^(&$%#$%(*&^%#$%&(01000Δ00011>?:"}%^$%#)(*&)%&#@!01111*&^(&$%#$%(*&^%#$%&(01101>?:"}%^$%#)(*&)%&#@!01001,.;]'<>:"{&(^*^#%$@%01110>?:"}%^$%#)(*&)%&#@!00111Δ01001*&^(&$%#$%(*&^%#$%&(01110Δ01000>?:"}%^$%#)(*&)%&#@!01001*&^(&$%#$%(*&^%#$%&(00111#$^%#)(^(^@>{($:"}>?01000Δ10010>?:"}%^$%#)(*&)%&#@!00101>?:"}%^$%#)(*&)%&#@!10001!@#$%^&*()(*&^%$%^&*10101#$^%#)(^(^@>{($:"}>?001001>?:"}%^$%#)(*&)%&#@!10011>?:"}%^$%#)(*&)%&#@!10100$%#^%&*^%$(*&^)#$%#@01001#$^%#)(^(^@>{($:"}>?01110>?:"}%^$%#)(*&)%&#@!00111Δ10100$%#^%&*^%$(*&^)#$%#@01000>?:"}%^$%#)(*&)%&#@!00101Δ00011$%#^%&*^%$(*&^)#$%#@01111>?:"}%^$%#)(*&)%&#@!10110*&^(&$%#$%(*&^%#$%&(00101,.;]'<>:"{&(^*^#%$@%10010Δ00110,.;]'<>:"{&(^*^#%$@%01001!@#$%^&*()(*&^%$%^&*10010>?:"}%^$%#)(*&)%&#@!00101Δ01111,.;]'<>:"{&(^*^#%$@%01110Δ01101>?:"}%^$%#)(*&)%&#@!11001Δ10010,.;]'<>:"{&(^*^#%$@%00101$%#^%&*^%$(*&^)#$%#@10100!@#$%^&*()(*&^%$%^&*10010!@#$%^&*()(*&^%$%^&*01001>?:"}%^$%#)(*&)%&#@!10110#$^%#)(^(^@>{($:"}>?00101>?:"}%^$%#)(*&)%&#@!ΔΔΔΔΔΔΔΔΔΔΔ

Sample output:

Tombstone three in bound from the north coming in high requesting the cover fire on my retrieve

IV. CONCLUSION

The encryption and decryption scheme designed by using computer theory machine is a lot easier to design and also the powerful encryption can be generated by using these tools which are difficult to decrypt but very easy to encrypt by using a certain scheme which is used in several ways, as we have shown a few here but its use is endless because now a days the communication media is advancing the sensitive information is transferred very often and these type of encryption can help us secure our data because security is very important in the communication between two parties because the sensitive information cannot be compromised at any cost.

REFERENCES

[1] Akanksha Mathur,"An ASCII value based data encryption algorithm and it's Comparison with other symmetric data encryption algorithms", ISSN: 0975-3397 Vol. 4 No. 09 Sep 2012.

[2] Vishwa gupta, "Advance cryptography algorithm for improving data security," Volume 2, Issue 1, January 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.

[3] Sunita Bhati, "A New Approach towards Encryption Schemes:Byte – Rotation Encryption Algorithm," Proceedings of the World Congress on Engineering and Computer Science, Vol. 2, WCECS-2012, October 24-26, 2012, San Francisco, USA.

[4] Taizo Anan, "*Paper encryption technology*," unpublished.

[5] Omar Chaudhary, "Infiltrate the Vault: Security Analysis and Decryption of Lion Full Disk Encryption in University of Cambridge, unpublished.

[6] Ravindu Madanayake, "Advanced Encryption Algorithm Using Fuzzy Logic in *IPCSIT vol. 27 (2012) © (2012) IACSIT Press, Singapore*.

[7] Majdi Al-qdah, Simple Encryption/Decryption Application, *Cyberjaya, 63100, Malaysia*, unpublished.