Cryptic Mining in Light of Artificial Intelligence

Shaligram Prajapat
Maulana Azad National Institute of Technology
Bhopal, India

Kajol Maheshwari
International Institute of Professional Studies
Devi Ahilya University, DAVV
Indore, India

Aditi Thakur
International Institute of Professional Studies
Devi Ahilya University, DAVV
Indore, India

Ramjeevan Singh Thakur
Maulana Azad National Institute of Technology
Bhopal, India

***Abstract*—*“The analysis of cryptic text is hard problem”*, and there is no fixed algorithm for generating plain-text from cipher text. Human brains do this intelligently. The intelligent cryptic analysis process needs learning algorithms, co-operative effort of cryptanalyst and mechanism of knowledge based inference engine. This information of knowledge base will be useful for mining data(plain-text, key or cipher text plain-text relationships), classification of cipher text based on enciphering algorithms, key length or any other desirable parameters, clustering of cipher text based on similarity and extracting association rules for identifying weaknesses of cryptic algorithms. This categorization will be useful for placing given cipher text into a specific category or solving difficult level of cipher text-plain text conversion process. This paper elucidates cipher text-plain text process first than utilizes it to create a framework for AI-enabled-Cryptanalysis system. The process demonstrated in this paper attempts to analyze captured cipher from scratch. The system design elements presented in the paper gives all hints and guidelines for development of AI enabled Cryptic analysis tool.**

Keywords*—*Cipher text; Cryptic analysis; Encryption algorithm; Artificial Intelligence (AI)

I. Introduction

Originally data mining techniques are concerned with information extraction at application level or for business and commercial need of individual or organization. The term "Cryptic-Mining" is used for low level information domain. This knowledge area increases the security level of information and power of cryptic algorithms by helping cryptanalyst. In order to strengthen the cryptosystem, automated tools can be developed that intelligently exploits patterns among cipher-text, plain-text, key size, key life time and log of partially recovered plain-text-cipher text derived knowledge. Cryptic mining domain assumes that cipher texts present in the network or stored encrypted files/logs are not 100% random and exhibits some patterns. These patterns may be useful to exploit weakness using mining algorithms.

Imagine the perspective of a cryptanalyst, who is interested to know about the type of enciphering algorithm. He is also interested in obtaining the plain text from encrypted text by exploiting patters or weakness. The obvious way to deal these intractable situations is mimic different theoretical and lengthy approaches by a human mind.

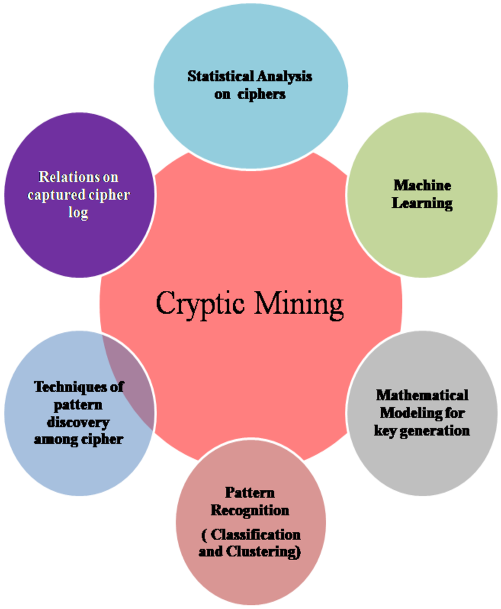
Statistical Analysis on ciphers
Relations on captured cipher log
Machine Learning
Cryptic Mining
Techniques of pattern discovery among cipher
Mathematical Modeling for key generation
Pattern Recognition (Classification and Clustering)

Fig. 1. Components of Cryptic Mining

Other alternative is to use AI and computational intelligence techniques that solves similar problems. In subsequent sections of this research work, a framework for AI enabled cryptic analysis system has been presented. This performs the cipher detection and successful conversion into plain-text in efficient way. This AI enabled system would help us to understand and analyze the various problems of cryptanalysis excluding strength and weaknesses of cryptic algorithms. This system would accept cipher texts generated from some algorithms and would try to extract meaningful information using some novel model or frameworks. Elucidation of cipher text-plain-text process has been shown on substitution cipher, such manner will resembles with the human way approach to solve the same problem. Later this concept would be generalized.

The flow diagram for schema of AI-enabled Cryptosystem has been depicted in the fig.3.It accepts a given cipher text (Substitution cipher), and attempts to transform it back to corresponding plaintext using process similar to human experts.

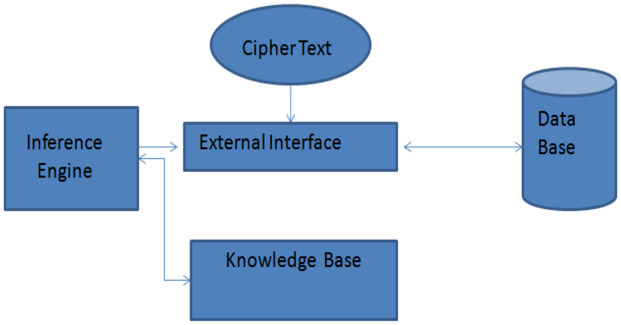

Fig. 2. Components of AI-Enabled cipher text to plain text conversion

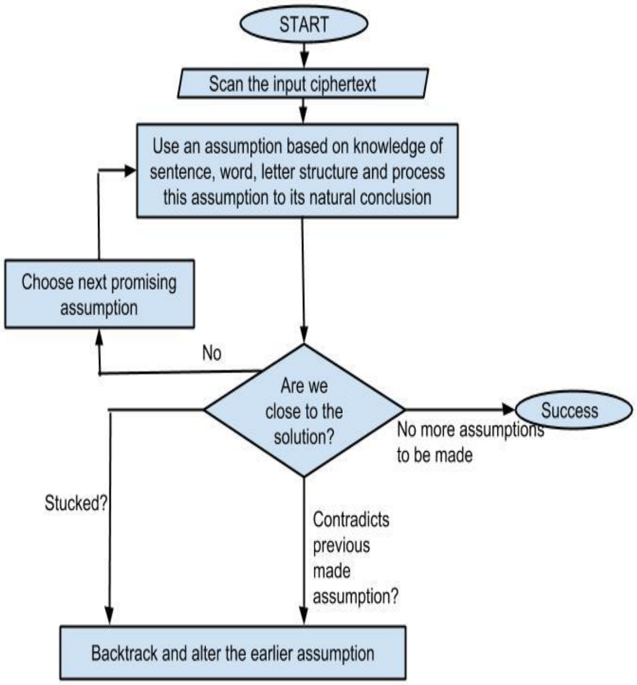
START
Scan the input ciphertext
Use an assumption based on knowledge of sentence, word, letter structure and process this assumption to its natural conclusion
Choose next promising assumption
No
Are we close to the solution?
Success
No more assumptions to be made
Stucked?
Contradicts previous made assumption?
Backtrack and alter the earlier assumption

Fig. 3. Schematic flow of AI-enabled cipher analysis system

In current typical cryptanalysis process, we limit ourselves to single substitution ciphers and we focus around "Transformation of cryptogram (cipher text) into message (plaintext) and vice-versa using single substitution cipher". In order to develop a cipher analysis system that transforms the cipher text into plaintext following steps are important: (1) Implementation of Cryptographic algorithms for producing substitution ciphertext. (2) Formulating the process of cryptanalysis. (3) Development of framework of AI-Enabled-cipher analysis system. (4) Implementation of framework for substitution ciphertext. (5) Extending the idea for categorization of cipher text generated from various symmetric key based cryptic algorithm (such as AES, DES, RC4, Blowfish and two Fish) (6) Evaluation of space and time complexity of new system.

In subsequent sections of this paper, we will describes the analysis of research topic using different examples and chalk down the system design based upon the proposed conceptual framework to be built. It includes various class diagrams and data flow diagrams describing the "dashboard". Further, system testing also has been discussed for using different examples to check functioning of each module. At the end future enhancements and new directions for further research work has been discussed in detail.

II. Basic Terminalogies

Cryptogram: A segment (word) of cipher text of length 1...n

Cryptographic Algorithms: The procedure that transforms messages (or plain-text) into cryptograms (or cipher text) and vice-versa.

Key Space: The set of possible keys K is called the key-space.

Substitution Cipher: It is the method of encoding by which units of plain-text are replaced with some other text.

Intractable Problem: Theoretically a solvable problem that takes too long time, in practice, for their providing useful solutions (e.g. deciphering cryptograms). Different alphabets are used in order to better distinguish plaintext and ciphertext, respectively. In fact these alphabets are the same.

A **cryptosystem** "S" can be defined by a 7-tuple:

S = (M, C, K_d, K_e, F, E, D) where:

M = Set of all possible **plaintext** m i.e. M= {m_1, m_2.......}. Each message m_i is the text to be encrypted (plaintext) and usually written in the lowercase alphabet: M = {a,b,c… x,y,z}.

C = Set of all possible **cipher text** c i.e. C = {c_1,c_2.......}.Each encrypted message (cipher text) c_i is usually written in uppercase alphabet: C = {A, B, C… X, Y, Z}.

K_d= Set of all possible **decryption key k** i.e. K_d = { k_1,k_2,....}

K_e=Set of all possible **encryption key k**' i.e. K_d= { k_1 ',k_2 ' ...}

F: K_d →K_e is a mapping from decryption key with corresponding encryption key. For Symmetric Cryptosystem **Kd** = **K_e** and F=I where Encryption and Decryption keys are same.

E is the relation E: K_e →(M→C) that maps encrypting keys k_e into encrypting relations e_{ke}: M→C. Each e_{ke} must be total and invertible, but need not be a deterministic function or onto.

D: K→(C→M) is the mapping that maps decrypting keys k into decrypting functions d_k: C→M. Each d_k must be a deterministic function and onto. E and D are related in that

K_e = F (k) ⊂ D (k) =d_k = e_{ke}^{-1} = E $(k_e)^{-1}$ m = $D_{[k]}$ (E $_{[F (k)]}$ (M)) Often e_{ke} are one to one and onto.

III. Review of Literature

In [1], a cryptosystem has been presented that records cipher generated using information recording techniques. Then, features from this information can be extracted to distinguish one cipher from others. Also, these features can be used to transform from future information into cipher-text.

In [2] analysis of cipher text was presented by combed algorithms simultaneously to transform cipher-text into plaintext information and addressed some problems like:{Block Length detection, stream detection, entropy analysis, recurrence analysis, dictionary based analysis, decision tree based problems}.

In [3], pattern recognition based enciphering algorithms have been presented for the identification of patterns using different classification techniques like:{ SVM, Naive Bayesian , ANN, Instance based learning , Bagging , AdaBoostM1, Rotation Forest, and Decision Tree }. It can be noted that, these approaches requires improvement in accuracy with increase in number of encryption keys.

In [4], some methods have been presented with application of tools like support vector machine to identify block-ciphers for different inputs. The first one works on cipher text and second method takes partially decrypted text derived from a cipher text as input. The SVM based method performs regression using hetero-association model to derive the partially decrypted text.

Nuhn and Knight [5], worked towards automation of deciphering of ciphers. They have analyzed large number of encrypted messages found from libraries and archives, and trained by human effort only by a small and potentially interesting subset. Their work attempts to reduce human effort as well as error in decryption. Also they were interested to develop a distinguisher (first trained and then predict) to know which enciphering method has been used to generate a given cipher text.

In [6], ANN based tool has been used for decoding of a ciphertext by a pattern classification problem.

A survey of AI techniques for development of cipher analysis has been demonstrated in [7], here main objective was to investigate usage of advanced AI techniques in cryptography and they found that AI based security measures can be developed but their performance will depends on the data representation and problem formulation.

In [8], Deciphering of messages from encrypted one using genetic algorithm has been presented. It searches the key space in encrypted text. They identified limitation that it didn't work with a two rotor problem in times comparable to those obtained using the iterative technique.

Frequency analysis in cipher-text provides a significant direction to cryptanalyst. According to Ragheb Toemeh and colleague in [9], this frequency analysis technique is used for framing objective function of cryptography. They studied the applicability of other methods like genetic algorithms for searching the key space of encryption scheme and presented cryptanalysis of polyalphabetic by applying Genetic algorithm.

Another survey based on parameters like queries, heuristics, erroneous information, group key exchange, synaptic depths has been conducted in [10], by Chakraborty and team . These parameters are suggested to improve the time complexity of algorithmic interception or decoding of the key during exchange.

In [11], A mathematical black-box model was proposed by Alallayah, AbdElwahed and Alhamami that builds the foundation for the development of Neuro-Identifier for determining the key from any given plain text-Cipher text pair. Some system identification techniques were combined with adaptive system techniques were used for the creation of the model.

All the above works and techniques follow in the direction of established long-fixed key sized algorithms. These algorithms rely on the ciphers would be secure enough if they are generated with keys of longer size. But in literature there are ciphers being generated through keys of short-fixed-length keys[12,13] varying with session to sessions. Ciphers generated through these AVK mechanism [14,15] are to be converted back into plain text.

IV. Experimental Design

For designing experimental setup it is necessary to first understand the complete mechanism of how the cipher analysis process works? How cryptanalysis applies rules of English grammar?

For this various grammar rules will be applied on the given cryptogram at different stages for each replacement which will aid in obtaining the desired plain-text.

Given following examples will be used to develop design model. Let us assume that cryptanalyst has captured following cryptogram: "*q azws dssc kas dxznn dasnn*". Now cryptanalyst may process according to following steps:

1) *To develop a model we take a hypothesis of solving a plain-text [Table 1]with one initial seed point .[Hint : wv]*

2) *Secondly the sentence is searched for smallest word (word with least number of letters), which in this case is the one-letter word 'q'. This word is replaced by plain letter 'A' as it has the highest priority for one-letter word according to the English grammar.*

3) *Next the first occurrence of double letter is searched in the sentence which is 'ss'. As it is in the middle of consonants, therefore it has to be a vowel according to English grammar and 's' is replaced by plain letter 'E' which has highest priority in this case.*

4) *Further the next smallest word is searched which is 'kea'. With this pattern the word with highest priority is 'THE'. Hence 'k' and 'an' are replaced by 'T' and 'H' respectively.*

5) *Now the word having the maximum number of letters replaced is 'HzVE' which can possibly be 'HIVE'('have' cannot be taken as 'A' is already used). Therefore 'z' is replaced with plain letter 'I'.*

6) *Next word 'dEEc' can be 'SEEN','BEEN','FEEL' etc. This word will be a verb, so we replace this word with 'SEEN'.*

7) *Now our sentence includes 'A HIVE SEEN', which is not possible as a hive cannot see. This states that we have possibly made some mistake with our assumptions before. Backtracking to the first assumption which was qa and*

changing qi to correct the sentence. Also the assumption zi has to be changed to za.

8) Further in the next word ‘SxAnn’, the double letter ‘nn’ will be a consonant according to the English language. Therefore ‘n’ is replaced by plain letter ‘L’ which has the highest possibility in this case.

9) Now ‘SxALL’ can possibly be ‘SMALL’ or ‘SHALL’. But observing the sentence structure it can be a noun or an adjective so ‘SMALL’ is used. Hence ‘x’ is replaced by plain letter ‘M’.

10)Finally we obtain the plaintext from the cryptogram given.

The above process can be summarized in Table1:

TABLE I. CRYPTANALYSIS STEPS WITH KNOWLEDGE SOURCE USED INTERFERENCE

Sno	**Cryptogram**	**Inference**	**Knowledge Source**	**Reference/ Remark**
1.	q azws dssc kas dxznn dasnn	wv	using hint /KS=direct substitution	
2	q azVs dssc kas dxznn dasnn	qa	KS=small word (n-gram :n=1)	
3	A azVs dssc kas dxznn dasnn	se,	KS=double letter	
4	A azVE dEEc kaE dxznn daEnn	kt, ah	KS=small word (n-gram: n=3)	
5	A HzVE dEEc THE dxznn dHEnn	zi	pattern matching (valid small word dictionary)	Dictionary
6	A HIVE dEEc THE dxInn dHEnn	ds, cn	pattern matching ,valid smallworld dictionary, sentence structure (position of word)	KS=Patterns
7	A HIVE SEEN THE SxInn SHEnn	qi, za	Sentence structure , KS=IsSolved	Backtracking
8	I HAVE SEEN THE SxAnn SHEnn	nl	KS=Double letter, KS=word structure	
9	I HAVE SEEN THE SxALL SHELL	xm	KS=word structure, pattern matching,KS=Sentence structures	
10	I HAVE SEEN THE SMALL SHELL		KS=IsSolved	

V. EXPERIMENTAL FINDING

It can be observed that a central place (like Dashboard) is needed to apply sources of knowledge. It would be useful to align with the assumptions made and to reason the consequences. Knowledgebase (a Data structure) KS will maintain log of many different sources of knowledge such as: Knowledge about grammar, spelling and vowels. At some point of time, specialization process (moving down) is followed (General to specific) during the replace of cryptogram with n=3 and ending with “e”. (for THE) and at some other points, Generalization process i.e. moving Up process is followed (from Specific to General) during the processing of cryptogram with n=4 and having pattern “?ee?”Which may be from {deer, beer, seen} but at the third position the word must be a verb instead of a noun, so “seen” should be final choice.

VI. FLOW DIAGRAM

In order to build a system flow of information from one component of system to other is depicted by fig.4, fig.5 and fig.6.

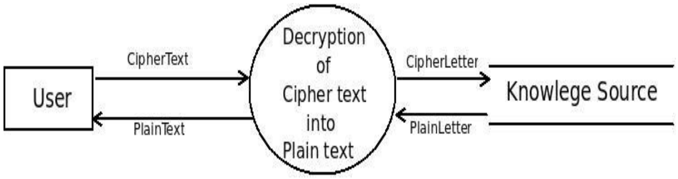

Fig. 4. Context flow diagram

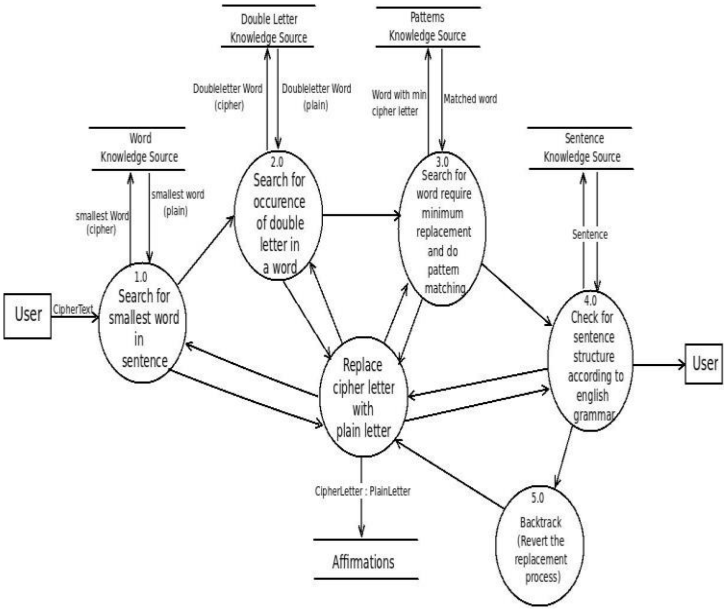
Double Letter Knowledge Source
Patterns Knowledge Source
Doubleletter Word (cipher)
Doubleletter Word (plain)
Word with min cipher letter
Matched word
Word Knowledge Source
Sentence Knowledge Source
smallest Word (cipher)
smallest word (plain)
2.0 Search for occurence of double letter in a word
3.0 Search for word require minimum replacement and do pattern matching
Sentence
User
CipherText
1.0 Search for smallest word in sentence
Replace cipher letter with plain letter
4.0 Check for sentence structure according to english grammar
User
CipherLetter : PlainLetter
Affimations
5.0 Backtrack (Revert the replacement process)

Fig. 5. First level data flow diagram

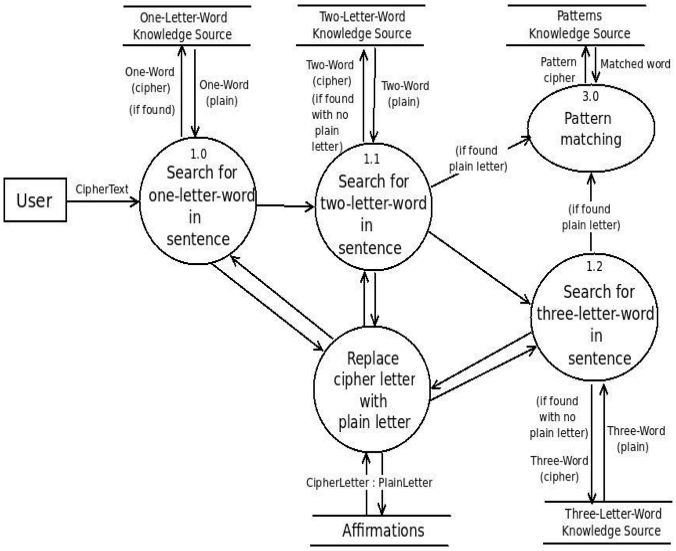
One-Letter-Word Knowledge Source
Two-Letter-Word Knowledge Source
Patterns Knowledge Source
One-Word (cipher) (if found)
One-Word (plain)
Two-Word (cipher) (if found with no plain letter)
Two-Word (plain)
Pattern cipher
Matched word
3.0 Pattern matching
User
CipherText
1.0 Search for one-letter-word in sentence
1.1 Search for two-letter-word in sentence
(if found plain letter)
(if found plain letter)
1.2 Search for three-letter-word in sentence
Replace cipher letter with plain letter
(if found with no plain letter)
Three-Word (cipher)
Three-Word (plain)
CipherLetter : PlainLetter
Affirmations
Three-Letter-Word Knowledge Source

Fig. 6. Second level data flow diagram

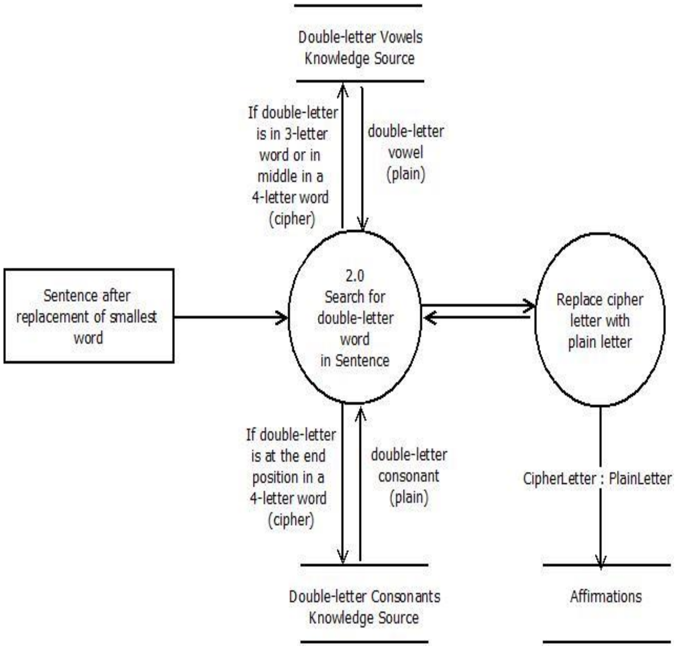
Double-letter Vowels Knowledge Source
If double-letter is in 3-letter word or in middle in a 4-letter word (cipher)
double-letter vowel (plain)
Sentence after replacement of smallest word
2.0 Search for double-letter word in Sentence
Replace cipher letter with plain letter
If double-letter is at the end position in a 4-letter word (cipher)
double-letter consonant (plain)
CipherLetter : PlainLetter
Double-letter Consonants Knowledge Source
Affirmations

Fig. 7. 2-Level Data flow diagram for process 2.0

VII. Modular Structure

For implementation of cryptosystem and cryptanalysis of substitution different cipher function structures are described below:

function1-def spell_check(word)

This module checks the spelling of the word and returns true if the spelling is correct.

function 2-def replacefunc(word, file_word)

This module replaces the word with a word from file and adds the entry in assumption(dictionary containing cipher Letter-plain, Letter pair)

function 3.-def transposition( )

This function displaces the cipher letter with plain letter according to the displacement in the plain letter with its corresponding cipher letter (key) in the assumption (dictionary). If the words replaced don't have correct spelling then the transposition is reverted back and the plain letters are again replaced with corresponding cipher letters which were added to assumption dictionary.

function 4.-def backtrack(word)

If no pattern match is found for a word then that word is passed as the argument to backtrack, it will replace the plain letter with their corresponding original cipher letter as the #assumptions made before was not correct

function 5.-def trans_status( )

After doing transposition it checks whether the transposition made was correct or not.

function 6.-def revert_trans( )

If the transposition made was correct then it displays the final sentence otherwise revert all the #changes made during transposition process

function 7.-def pat_rep(lst, fil, cnt)

pat_rep function replaces the words from list with suitable word from file according to condition. It has three arguments:

lst**:** list of specific words(i.e 2-letter, 3-letter etc) if the sentence containing cipher.

fil: text file of containing 2-letter-letter etc plain-letter words corresponding to list.

cnt: counter to mention the position in the file

function 8.-def pattern(word, fil, cnt)

If the word contains one or more plain letter ***pattern function*** matches the word with every word in file and replaces if a pattern is matched. It has 3 arguments:

word: word from sentence containing a capital letter

fil: corresponding file(for ex: 4_word file for 4-letter word)

cnt: counter that mentions position in the file

function 9.-def double_letter(word)

This function checks if a word (input) contains any double letter, if yes it replaces the double letter cipher with appropriate plain letter according to its position (i.e. if in middle it will be a vowel and if end it will be a consonant according to English grammar rules)

function 10.-def one_letter( )

If the sentence contains ***one-letter-word*** in cipher then this function will replace that cipher with the possible plain one-letter-word and will make entry according to the assumption.

function11-def find_key(value)

This function finds the corresponding ***cipher(key)*** letter of the plain ***letter(value)*** given as argument from the dictionary "assumption"

VIII. Test Case Development

Test cases are developed to validate and verify the working of system in two situations. Case-1 and Case-2.

Case-1: For testing transposition

Let sentence given by user: sent_1 = “k co c iktn”

Case-2: For testing english grammar

Input Sentence supplied by user:

sent = “dwer er ed”

TABLE II. Steps Followed for Case 1

S.no	Module name	Test Cases	Result	Response
1.	Enter valid cipher sentence	Check chars of sent	Returns true if the sentence contains only alphabets otherwise false	OK
2.	**one_letter()**		Replace one-letter cipher word with the plain word chosen from the file containing one-letter words	OK
3.	one_letter() sent=”k co c iktn”	action performed on sent	sent=”I Ao A iItn”	OK
4.	**transposition()**		Finds the difference between the replaced cipher letter and its corresponding plain letter and replace remaining cipher letter with plain letter with same difference	OK

5.	transposition()
	sent = “I Ao A iItn”
		action performed on sent	sent = “I AM A GIRL”	OK

6.	**spell_check (word)**		Returns true if correct spelling else false	OK
6.1	spell_check (“I”)	correct spelled word	Returns true	OK
6.2.	spell_check (“AM”)	correct spelled word	Returns true	OK
6.3.	spell_check (“A”)	correct spelled word	Returns true	OK
6.4.	spell_check (“GIRL”)	correct spelled word	Returns true	OK
7.	**trans_status()**		Returns false if spelling of any of the word in sent is wrong else true	OK

8.	trans_status()
	sent=”I AM A GIRL”
		all words are correct spelled	returns true	OK

TABLE III. Steps Followed for Case 2

S.no	Module name	Test Cases	Result	Response
1.	main()		Calls all functions according to condition	OK
2.	pat_rep(lst,fil,cnt)		Search the word from list from file according to conditions met	OK
3.	replacefunc(word,file_word)		Replace each chars of word with the corresponding chars of file_word and made the entry of pair(cipherletter:plainletter) in the dictionary ‘assumption’	OK
4.	pattern(word,fil,cnt)		Search the word from list according to the pattern formed	OK
5.	backtrack(word)		Reverts back the previous assumptions made if pattern is not found for *word* i.e replace the plain text with their original ciphertext in sent	OK
6.	check_sent(sent)		Checks the grammar of sentence and returns true if correct else false	OK
7.	transposition()		Finds the difference between the replaced cipher letter and its corresponding plain letter and replace remaining cipher letter with plain letter with same difference	OK
8.	Enter valid cipher sentence	Check each characters of sent	Returns true if the sentence contains only alphabets otherwise false	OK
9.	main() started			
10.	sent = “dwer er ed”	check for smallest word	Two-letter word found	OK

11.	pat_rep(two_w,tw,cnt2)
	two_w = [er,ed]
		Actions performed on the words of list two_w,hence on sent	Replaces ‘er’ with ‘OF’(first word in fil tw)
			sent = “dwOF OF Od”
				OK

12.	replacefunc(er,OF)	Replacement done on sent	Replaces ‘er’ with ‘OF’(first word in fil tw) sent = “dwOF OF Od”	OK

			assumption={‘e’:’O’ , ’r’:’F’}	
13.	pattern(Od,etw,e2cnt) etw: file containing 2-letter words at ending position of sent	Search matched word	No pattern found for pattern=’O.’	OK
14.	backtrack(Od)	Action performed on sent and assumption	Replaced all the plain letter i.e. F and O with corresponding cipher letter from assumption and calls main() again	OK
15.	main() called			
16.	sent = “dwer er ed”	Check for smallest word	Two-letter word found	OK

16.1	pat_rep(two_w,tw, cnt2)
	two_w = [er, ed]
			Start search for the word in “tw” according to condition after the word last searched	OK

16.1.1	replacefunc(er,TO)	Replacement done on sent	Replaces ‘er’ with ‘TO’
			sent = “dwOF TO Od”
			assumption={‘e’:’T’ , ’r’:’O’}
				OK

16.1.2	pattern(Td,etw,e2cnt) etw: file containing 2-letter words at ending position of sent	Search matched word	No pattern found for pattern=’T.’	OK
16.1.2.1	backtrack(Td)	Action performed on sent and assumption	Replaced all the plain letter i.e. T and O with corresponding cipher letter from assumption and calls main() again	OK
17.	main() called			
18.1	sent = “dwer er ed”	Check for smallest word	Two-letter word found	OK

19.	pat_rep(two_w,tw, cnt2)
	two_w = [er, ed]
			Start search for the word in “tw” according to condition after the word last searched	OK

19.1.1	replacefunc(er,IS)	Replacement done on sent	Replaces ‘er’ with ‘IS’
			sent = “dwIS IS Id”
			assumption={‘e’:’I’ , ’r’:’S’}
				OK

19.1.2	pattern(Id,etw,e2cnt)
	etw: file containing 2-letter words at ending position of sent
		Search matched word	Match found for pattern=’I.’
			match = “IT”
			sent = “TwIS IS IT”
			assumption={‘e’:’I’ , ’r’:’S’, ‘d’:T}
				OK

20	return to main()			
21	transposition()	Check difference between cipher letter and plain letter	Difference is not same, therefore returns False	
22	sent = “TwIS IS IT”	Check for word having length greater than two	Four-letter word found	OK

22.1	pat_rep(four_w,fw, cnt4)
	four_w = [TwIS]
			As word contains plain letter so calls pattern()	OK

22.2	pattern(TwIS,sfw,s4cnt)
	sfw: file containing 4-letter words at starting position of sent
		Search matched word	Match found for pattern=’T.IS’
			match = “THIS”
			sent = “THIS IS IT”
			assumption={‘e’:’I’ , ’r’:’S’, ‘d’:’T’, ‘w’:’H’}
				OK

sent = “THIS IS IT” and assumption={‘e’:’I’ , ’r’:’S’, ‘d’:’T’, ‘w’:’H’}				
23	check_sent(sent)	Check grammar of sent	Returns true	

IX. Conclusion

This paper is an attempt to demonstrate the demonstrate cipher text -plain text conversion process for analysis of cryptic text. AI has been used to get the feasible solution of hard problem. By generalizing the conversion process system for obtaining plain-text from input cipher text is the central objective. The developed system would analyze and learn for pruning. This paper has demonstrated cipher text-plain text process completely and created a framework for AI-enabled-Cryptanalysis system, Data Flow Diagrams and appropriate test cases. This schema and plan would be suitable for development of AI enabled Cryptic analysis tool and in turn they will evaluate strength of any cryptosystem.

X. Future Enhancement

AI-based-crypto system works correctly for the basic cipher-text to plain-text conversion process. To extend this further to fulfill various requirements following enhancements are suggested.

1) Current work can be extended to incorporate ciphers other than substitution and transposition cipher. That is, present system response is fine for transposition cipher and substitution cipher, but cipher types are more than two. This will require testing with different algorithm, method and cipher text. So that extended version is fit and deciphers it accordingly.

2) Incorporating plain-text of multiple languages in the process is also desirable. That is, current elucidation demonstrated in this work deciphers and outputs result in English. Maximum number of ciphers gives English plain-text on decryption. But over the communication channel languages local, non-English languages are also exchanged. For decryption of cipher text yielding other language plain text, the grammar rules of that particular language has to be applied.

3) Extension of character set with adding special characters and symbols will make the current system more flexible. The reason behind this is, day-by-day increasing amount of data transferred, and the need to encrypt it in a more complex way is mandatory for securing information from unauthorized users. Hence special characters and numbers are used to generate a more complex cipher patterns. Deciphering these ciphers using algorithm with condition for checking these symbols together with the English alphabets will be necessary.

4) Extension for n-gram (n>4) will increase the power of cipher analysis. That is checking cipher words with having length more than 4 and words which are not present in any knowledge source, needs to be worked out. Currently the Knowledge source, include files having upto 4-letter words. More generalized approach is needed for words having length more than 4. This may require a tool for checking the spellings of every possible word which states that the spelling is correct or not.

ACKNOWLEDGMENT

This work is supported by research project under Fast Track Scheme for Young Scientist from DST, New Delhi, India. Scheme 2011-12, No. SR/FTP/ETA-121/ 2011 (SERB), dated 18/12/2012.

REFERENCES

[1] Khadivi P, Momtazpour M. “Ciphertext classification using data mining”,International Symposium on Advanced Networks and Telecommunication Systems IEEE- ANTS, pp.64-66, 2010.

[2] Shivendra Mishra, Dr. Aniruddha Bhattacharya , “Pattern analysis of cipher text : a combined approach”, proceeding of Recent Trends in Information Technology (ICRTIT), 2013 International Conference on ,25-27 July 2013,393 - 398, DOI:10.1109/ICRTIT.2013.6844236.

[3] Sushila Omer Sharif, saad P. Mansoor , “Performance evaluation of classifiers of encryption algorithm”, ACEEE International Journal on Network Security , Vol. 02, No. 04, Oct 2011.

[4] S. Swapna, A. D. Dileep, C. Chandra Sekhar, and Sri Kant, “Block cipher identification using support vector classification and regression,” Journal of Discrete Mathematical Sciences and Cryptography, vol. 13, no. 4, pp. 305- 318, August 2010.

[5] Malte Nuhn, Kevin Knight , “Cipher type detection”, http://www-i6.informatik.rwth-aachen.de/~nuhn/2014-classification-poster.pdf, pp.1769–1773,2014

[6] Sambasiva Rao Baragad, P. Satyanarayana Redd , “Studies on the advancements of Nerual Networks and Neural Network based cryptanalytic works”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),Volume 2, Issue 5, September – October 2013 ISSN 2278-6856

[7] E.C. Laskari, G.C. Meletio, Y.C. Stamatiou, M.N.Vrahatis ,“Cryptography and Cryptanalysis through Computational Intelligence”, proceeding of Computational Intelligence in Information Assurance and Security, Studies in Computational Intelligence Volume 57,2007, pp 1-49

[8] A.J. Bagnall, G.P. Mckeown, V.J. Rayward Smith , “Cryptanalysis of a three rotor machine using a genetic algorithm”, In proceedings of 7th International Conference on genetic algorithms,ICGA97,1997.

[9] Sandip Chakraborty, Jiban Dalal, Bikramjit Sarkar, Debaprasad Mukherjee , “Neural Synchronization based secret key exchange over public channels: A Survey”, Journal of Engineering Science and Technology Review 8(2) (2015) 152 – 156.

[10] Khalid Alallayah, Moamed Amin, Wail AbdElwahed, Alaa Ahamami ,“Applying Neural Networks for simplified data encryption standard (SDES) cipher system cryptanalysis”, Vol. 9, pp.163-169,2012.

[11] Shaligram Prajapat, A. jain, R.S.Thakur, “A Novel Approach For Information Security with Automatic Variable Key Using Fibonacci Q-Matrix”, International Journal of Computer & Communication Technology (IJCCT) ISSN (ONLINE): 2231 - 0371 ISSN (PRINT): 0975 – 7449 Vol-3, Iss-3, 2012, p.p. No. 54-57.

[12] Shaligram Prajapat, D. Rajput, R. S. Thakur, "Time variant approach towards Symmetric Key " ,Science and Information Conference 2013,October 7-9, 2013, London, UK p.p.398-405.,Technically Co-Sponsored by: IEEE Computer Society, UKRI Section, IEEE Computational Intelligence Society,UKRI Section, IEEE Consumer Electronics Society,, IEEE Sponsored and Organized by The Science and Information (SAI) Organization

[13] Shaligram Prajapat, R.S. Thakur, "Towards Optimum size of key for AVK based cryptosystem", Communicated and CJICT, Nigeria in June-Dec. 2015.ISSN (Online): 2354 - 3507; ISSN (Print): 2354 - 3566

[14] Shaligram Prajapat, R.S.Thakur, "Markov Analysis of AVK Approach of Symmetric Key Based Cryptosystem ",Computational Science and Its Applications, ICCSA 2015,Springer LNCS: Volume 9159, 2015, pp 164-176,Jun 2015,doi:10.1007/978-3-319-21413-9_12,ISBN:9783319214139 and 9783319214122.

[15] Shaligram Prajapat, R. S.Thakur, "Cryptic-Mining: Association Rules Extractions Using Session Log ", Computational Science and Its Applications, ICCSA 2015,Springer LNCS: Volume 9158, 2015, pp 699-711,Jun 2015,doi:10.1007/978-3-319-21413-9_12.

[16] Claudia Oliveira, J. A. Xex´eo, C. A. Carvalho,”Clustering and Categorization Applied to Cryptanalysis”,Taylor and Francis 2007

[17] M.F. Uddin and A.M. Youssef,Cryptanalysis of simple substitution ciphers using particle swarm optimization, Evolutionary Computation, 2006. CEC 2006.IEEE Congress on, 0-0 2006, pp. 677 -680.

[18] George Nagy, Sharad C. Seth and Kent Einspahr, “Decoding Substitution Ciphers by Means of Word Matching with Application to OCR”, 1987

[19] Amrapali Dhavare, Richard M. Low & Mark Stamp ,“Efficient Cryptanalysis of Homophonic Substitution Ciphers”, 2013

[20] Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young(Ph.D.), Jim Conallen, Kelli A. Houston, “Object-oriented Analysis and Design with applications”, Addison-wesley publishing company, Rational, Santa Clara, California 3rd Edition

[21] S. William and Stalling, Cryptography And Network Security, 4/E. Pearson Education India, 2006.

[22] http://www.nltk.org

[23] http://what-when-how.com/artificial-intelligence/automated-ryptanalysis-artificialintelligence/

[24] http://cse.ucdenver.edu/~rhilton/docs/Cryptanalysis-Against-Monosub-Ciphers.pdf

[25] http://people.csail.mit.edu/hasinoff/pubs/hasinoff-quipster-2003.pdf

[26] http://scottbryce.com/cryptograms/stats.htm

[27] http://jeremykun.com/2012/02/03/cryptanalysis-with-n-grams/