Proc. of The Second Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2015

ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-77

Cryptic Mining in Light of Artificial Intelligence

[Shaligram Prajapat , Aditi Thakur ,Kajol Maheshwari, Ramjeevan Singh Thakur]

***Abstract*—"Analysis of cipher text is intractable problem", for which there is no fixed algorithm. For intelligent cryptic analysis there would be a need of cooperative effort of cryptanalyst and inference engine .The information of knowledge base will be useful for mining tasks such as information about the classification of cipher text based on encrypting algorithms, clustering of cipher text based on similarity, extracting association rules for identifying weaknesses of cryptic algorithms. This categorization will be useful for placing given cipher text into a specific category of difficulty level of cipher text-plain text conversion. This paper attempts to create a framework for AI-enabled-Cryptanalysis system. The process depicted in the paper generalizes the idea for development of from scratch. The paper also presents useful system design diagrams for development of extended AI based Cryptic cipher analysis tool.**

Keywords— cipher text, cryptic analysis, encryption algorithm, Artificial Intelligence (AI)

I. Introduction

Imaging two situation from our neighborhood, these incidences exhibits different aspects of human behavior.

Situation-1: Consider a problem, where a researcher has to focus on a single speaker in a conference where several conversations are going on simultaneously.

Situation-2: A cryptanalyst is interested to identify the type of decryption algorithm used for obtaining the plain text from encrypted text.

Situation-1, will require the listener to distinguish the meaningful data and filter out all unwanted conversations. Whereas Situation-2, will involve different approaches according to the encrypted text. The obvious way to deal these intractable situations are treated with different theoretical and lengthy approaches by a human mind.

Using AI and Computational Intelligence, to solve similar problems an attempt has been made in this research work. We have made an attempt for development of an intelligent system that performs the cipher detection and clustering and categorization task in efficient way. This AI enabled system would help us to understand and analyze the various problems of cryptanalysis including 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.

Experimentation would be done initially on some specific type of ciphers e.g. substitution cipher, in such a manner will resemble with the way of approach of human to solve the same problem. Later on the concept would be generalized.

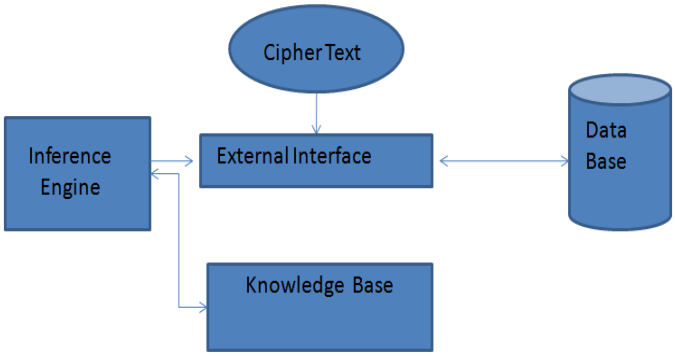

Fig 1: Scheme of AI-Enabled CryptoSystem

Research Problem central to this paper would be,"To develop an automated system that accepts a given cipher text , attempts to transform it back to original plaintext, using similar way as human experts does it otherwise. For simplicity the initially it will work for substitution cipher."

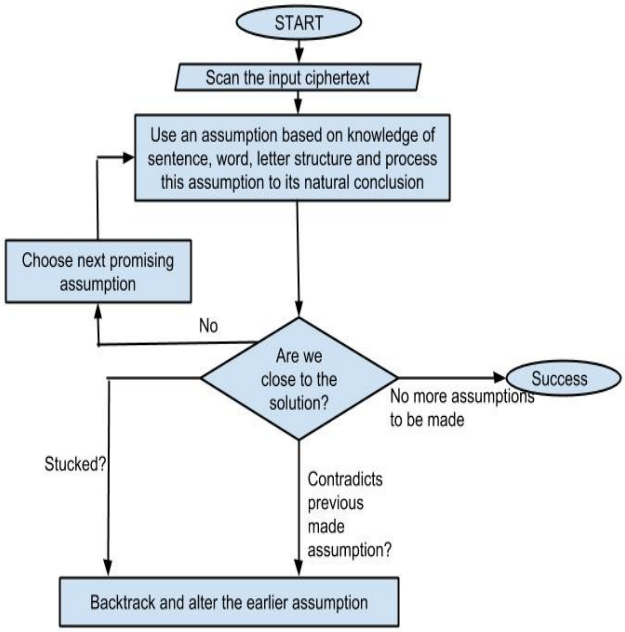

Fig.2 Schematic flow for implementation of AI-enabled Cryptosystem

To achieve this task, first we have to understand and design an automated system for **"Cryptanalysis".** In general it performs, deciphering analysis on cryptograms, in polynomial time by inventing sophisticated techniques. Our problem is

Shaligram Prajapat [1], Aditi Thakur[2], Kajol Maheshwari[3]
IIPS-DAVV,Indore
India
[1]shaligram.prajapat@gmail.com, [2]a.thakur73.at@gmail.com
[3]maheshwari.kajol@gmail.com

Ramjeevan Singh Thakur
MANIT,Bhopal
India
ramthakur2000@yahoo.com

Proc. of The Second Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2015

ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-77

relatively simple because we limit ourselves to single substitution ciphers. We can narrow down the problem domain as: "Transforming the cryptogram (cipher text) into message(plaintext) and vice-versa using single substitution cipher". In order to develop such cryptosystem that transforms the cipher text into plaintext using substitution cipher. This aim can be subdivided into following steps:(1) Implement Cryptographic algorithms for cipher generation: substitution cipher.(2) understand the process of cryptanalysis.(3)Develop and understand model and framework of AI-Enabled-cryptanalysis based system.(4)implement the model and framework for some specific ciphers: substitution cipher. (5)extend the idea for categorization of cipher text generated from different cryptographic techniques: such as "AES" , "DES", "RC4" ,"Blowfish", "twoFish" etc(6)analyze space and time complexity of the of newly developed 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 have been discussed for using different examples to check functioning of each module. At the end future enhancements and opens new directions for further research work has been discussed in detail.

II. Basic Terminologies

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}:MC. 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.

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

Cryptographic Algorithms: The procedure that transforms messages (or plaintext) 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 plaintext are replaced with some other text.

Intractable Problem: Theoretically solvable problems 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.

III. Experimental Design

For developing the system design it is necessary to first understand the complete mechanism of how the decryption process will be implemented. The system performs cryptanalysis on the basis of english grammar rules. 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 plaintext. 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*".

TABLE I. CRYPTANALYSIS STEPS WITH KNOWLEDGE SOURCE USED INFERENCE

Sno	Cryptogram	Inference	Knowledge Source	Reference/ Remark
1.	q azws dssc z dsz dasnn	w→V	using hint /KS=direct substitution	
2	q azVs dssc z dsz dasnn	q→A, z→I	KS=small word (n-gram :n=1)	
3	A aIVs dssc I dsI dasnn	s→E,	KS=double letter	
4	A aIVE dEEc I dEI daEnn	a→H	pattern matching (valid small word dictionary)	Dictionary
5	A HIVE dEEc I dEI dHEnn	d→S, c→N	pattern matching ,valid smallworld dictionary, sentence structure (position of word)	KS=Patterns
6	A HIVE SEEN I SEI SHEnn	q→I, z→A	Sentence structure , word spelling KS=IsSolved	Backtracking
7	I HAVE SEEN A SEA SHEnn	n→L	KS=Double letter, KS=word structure	
8	I HAVE SEEN A SEA SHELL		KS=IsSolved	

IV. Observation

From above table following points can be noted. A central place (like Dashboard) is used to apply sources of knowledge to the assumptions and to reason the consequences. Knowledge Data structure KS will use many different sources of knowledge such as: Knowledge about grammar, spelling

Proc. of The Second Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2015

ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-77

and vowels. At some point, specialization process (moving down)is followed (General to specific) during the replace of cryptogram with n=4 and having pattern ".IVE". (for HIVE) 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.

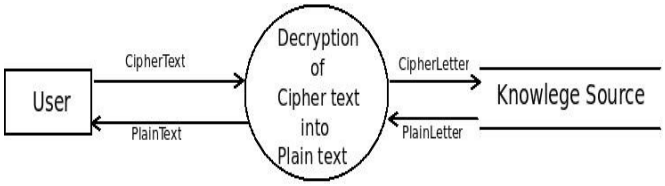

Fig 3. Contextual diagram

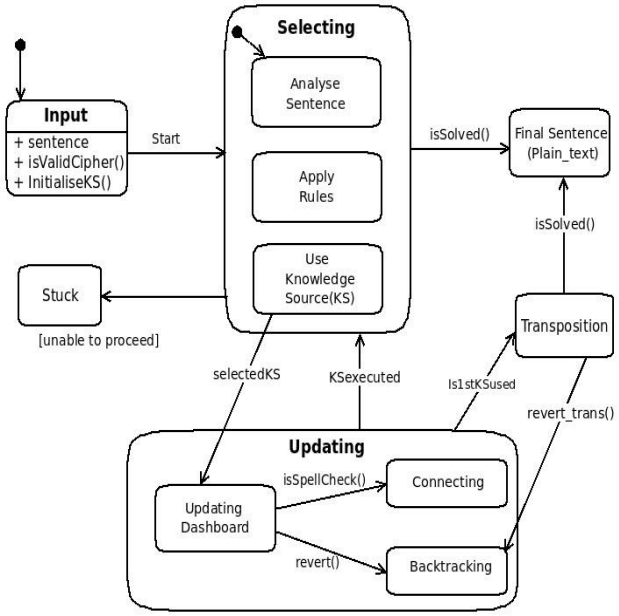

Fig 4 : Conceptual flow of the system

V. DataStructure for implementation

The implementation and code snippet of decryption technique mentioned above i.e. cryptanalysis of substitution cipher is described here

TABLE II. LIST OF FUNCTIONS

1.	def spell_gram_check(sent):	This module checks the spelling and grammar of the word and returns true if the spelling and grammar is correct.
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)
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.
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
5.	def trans_status():	After doing transposition it checks whether the transposition made was correct or not .
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
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
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
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)
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.
11.	def find_key(value):	This function finds the corresponding ***cipher(key)*** letter of the plain ***letter(value)*** given as argument from the dictionary "assumption"

VI. Test case Development

Sentence given by user:
sent = "q azws dssc z dsz dasnn"

TABLE III. SYSTEM TESTING WITH CONCLUSION

Proc. of The Second Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2015

ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-77

S.no	Module name	Test Cases	Result	Conclusion
	Enter valid cipher sentence	Check each characters of sent	Returns true if the sentence contains only alphabets otherwise false	OK
	main() started			
		Ask user for hint	Replace ‘w’ with ‘V’	OK
	sent = “q azVs dssc z dsz dasnn” assumption = {‘w’:’V’}	check for smallest word	one-letter words found	OK
1)	one_letter()	Actions performed on the 1-letter words hence on sent	replaces ‘q’ with ‘A’(higbest priority) and ‘z’ with ‘I’ sent = “A aIVs dssc I dsI dasnn” assumption = {‘w’:’V’,’q’:’A’:’z’:’I’}	OK
2)	sent = “A aIVs dssc I dsI dasnn”	check for double letter	double-letter in a word found	OK
2.1)	double_letter(‘dssc’)	Double-letter ‘ss’ found in ‘dssc’	Replaces ‘ss’ with ‘EE’ having highest priority for double letter vowel sent = “A aIVs dEEc I dsI dasnn” assumption = {‘w’:’V’,’q’:’A’,’z’:’I’,’s’:’E’}	OK
3)	sent =”A aIVE dEEc I dEI daEnn”	Finds the word having maximum letters replaced	Finds ‘aIVE’ and ‘dEEc’ and does pattern matching	OK
3.1)	pattern(aIVE,fw, cnt4) fw: file containing 4-letter words cnt4:counter in file fw	Search matched word	word found for pattern=’.IVE’ match = ‘HIVE’ sent =”A HIVE dEEc I dEI dHEnn” assumption = {‘w’:’V’,’q’:’A’,’z’:’I’,’s’:’E’,’a’:’H’}	OK
3.2)	pattern(dEEc,fw, cnt4) fw: file containing 4-letter words cnt4:counter in file fw	Search matched word	word found for pattern=’.EE.’ match = ‘SEEN’(highest priority) Replaces ‘d’ with ‘S’ and ‘c’ with ‘N’ sent =”A HIVE SEEN I SEI SHEnn” assumption = {‘w’:’V’,’q’:’A’,’z’:’I’,’s’:’E’,’a’:’H’,’d’:’S’,’c’:’N’}	OK
4)	sent =”A HIVE SEEN I SEI SHEnn”	Check the structure of sentence and spelling	Error in sentence structure as hive cannot see and spelling mistake in word ‘SEI’	OK
4.1)	Spell_gram_check(sent)	Returns False	Calls backtrack() function	OK
4.2)	backtrack(HIVE)	Reverts the first assumption	Removes the the entry ‘z’:’I’ and ‘q’:’A’ from assumption and replaces ‘z’ with ‘A’ and ‘q’ with ‘I’ in sent sent =”I HAVE SEEN A SEA SHEnn” assumption = {‘w’:’V’,’q’:’I’,’z’:’A’,’s’:’E’,’a’:’H’,’d’:’S’,’c’:’N’}	
5)	sent =”I HAVE SEEN A SEA SHEnn”	Checks the last word left in sent	Double letter found	
5.1)	double_letter(‘SHEnn’)	Double-letter ‘nn’ found in ‘SHEnn’	Replaces ‘nn’ with ‘LL’ having highest priority for double letter consonants sent =”I HAVE SEEN A SEA SHELL” assumption = {‘w’:’V’,’q’:’I’,’z’:’A’,’s’:’E’,’a’:’H’,’d’:’S’,’c’:’N’,’n’:’L’}	OK
sent = “I HAVE SEEN A SEA SHELL” and assumption={‘w’:’V’,’q’:’I’,’z’:’A’,’s’:’E’,’a’:’H’,’d’:’S’,’c’:’N’,’n’:’L’}				
6)	Spell_gram_check(sent)	check grammar of sent	returns true	

VII. Future Enhancement

AI-based crypto system has been implemented correctly for the basic requirements, In future changes can be made in order to fulfill various requirements as they occur. The system can be refined to provide more responsiveness, efficiency, reliability and user-friendly.

(1)Decryption for ciphers other than substitution and transposition cipher: The system responses works fine for transposition cipher and substitution cipher. But types of cipher does not limit to these two. Encryption can be done using various complex methods. Decryption for such methods can be implemented in order to make the system responsive for large number of inputs. This will require different

ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-77

algorithm for each method and the input will have to be tested for each method so as to determine where it fits and decrypt it accordingly.

(2)Decryption for languages other than English: Maximum number of ciphers gives English plaintext on decryption. But nowadays languages other than English are also used since transfer of data in these languages has started. For decryption of cipher text yielding other language plain text, the grammar rules of that particular language has to be applied.

(3)Decryption of text containing special characters and symbols : As the amount of data transferred is increasing day-by-day, 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. To decrypt these ciphers the algorithm should include condition for checking these symbols too along with the English alphabets used.

(4)Checking words having length more than 4 and words which are not present in any knowledge source : Currently the Knowledge sources used above, include files having upto 4-letter words with some limited number of words for each. A more generalized approach is needed for words having length more than 4. This will 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.The work is also supported by

References

[1] Claudia Oliveira, Jos´e Ant^onio Xex´eo, Carlos Andr´e Carvalho"*Clustering and Categorization Applied to Cryptanalysis*",Taylor and Francis 2007

[2] 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.

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

[4] *Efficient Cryptanalysis of Homophonic Substitution Ciphers* by Amrapali Dhavare, Richard M. Low & Mark Stamp , 2013

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

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

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

[8] 2.http://what-when-how.com/artificial-intelligence/automated-cryptanalysis-artificial

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

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

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

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

About Author (s)

Shaligram Prajapat, as received B.Sc.(Elex), M.Sc.(CS).,UGC.(NET), M. Tech.(CS),,M..Phil. (CS) from Devi Ahilya University Indore, He is associate professor and In-charge of Development Center at IIPS D. A. University Indore. With over 15 years of teaching experience of UG and PG courses, He has reviewed five international books of Pearson education, 10 papers in reputed conferences, international journals including Springer and Atlantis press. He has also presented paper in international and national conferences. He is member of various professional bodies like IEEE, ISTE, ACM, CSI, CSTA, IAENG, IEEE(Computer Society),IRED.

Aditi Thakur is pursuing Master of Technology M. Tech.with specialization in Information Technology(IT) from International Institute of Professional Studies(IIPS), Devi Ahilya University Indore. Her core area of interest includes Cryptography, linguistic analysis, Information security and Web Application development. She is a member of the PyChef, Development Center (DC) of IIPS-DAVV, India. She is python developer and participated and conducted many workshops. She is member of IRED.

Kajol Maheshwari is pursuing Master of Technology M.Tech. with specialization in Information Technology (IT) from International Institute of Professional Studies(IIPS), DAVV, Indore. Her core area of interest includes Cryptography, linguistic analysis, Information security and Web Application development. She is a member of the PyChef, Development Center (DC) of IIPS-DAVV, India. She is python developer and participated and conducted many workshops. She is member of IRED.

Dr. Rajeevan Singh Thakur is Associate Professor in MANIT, India. He is a Educationist, Researcher and Consultant in Computer Science and Information Technology. He earned MCA, M.Tech, Ph.D. (Comp.Sc.). He has published more than 75 Research Paper in National, International, Journals and Conferences. He has visited several Universities in USA, Hong Kong, Iran, China, Thailand, Malaysia, and Singapore. His areas of interest include Data Mining, Data Warehousing, Web Mining, Text Mining, and Natural Language Processing. He has received DST Young Scientist Award-2011 in Engineering under Fast Track Scheme, Department of Science & Technology, New Delhi, India.