Metric to determine language complexity using dictionary Method percentage retrieval

Devasish Pal, N.V. Ganapathi Raju' Gautam Pal

Abstract: ***For communication through computer network, previously only English language using ASCII mode was used. Subsequently when Unicode was introduced, computer communication became a possibility for all language texts. This aspect generated interest in the field of language processing. Various studies have been carried out on language processing and its complexity issues. Various metrics were used to determine language complexity such as lexical density, morphological density, semantics etc. but there was no consistency in results. A language which appears most complex using one metric does not appear the same using other metric. This paper introduces a new metric to determine the complexity of a language which is consistent and with proven results. It introduces the concept of network security where using dictionary method, the percentage retrieval of an encrypted text is calculated using an encryption algorithm, fixed length key, fixed corpus size etc. Lesser is the percentage retrieval, greater is the security and language complexity. Comparison has been made with the results on language complexity independently carried out on various Indian languages by the research scholars of Central University, Hyderabad based on Morphological and lexical density. Pattern observed on their eight Indian languages by the research scholars of Central University and the percentage retrieval on the same Indian languages in my work are identical which proves my work. Hence it can be concluded that lesser is the percentage retrieval, security increases for the sample text data considered and proportionately the complexity of that particular language increases Sample data encryption has been carried out using substitution method.***

Index Terms: language complexity, dictionary file, coded file, morphology, lexical, percentage retrieval

I. INTRODUCTION

Popularity and development in the field of computer networks and internet with free access to it by all users throughout the world has become an area of security concern. Security of data while transit across the network and information stored in individual computer systems are two separate issues of network security. Further data is divided as text data and non-text data. Before Unicode was introduced English language was used for text data. Localization process and introduction to Unicode [2] encouraged text data and information being transmitted in various languages throughout the internet. This aspect generated interest in the field of language processing. Vast research has been carried on language processing to determine language complexity issues. Some of the parameters which have been considered as metric to study complexity of a language are: Morphology, Phonology, Pragmatics, Grammar, Semantics, Lexical density, total characters of a language, Redundancy, Syntax etc.

This paper is dealing with text data security during transit through the network and links this aspect to introduce a new metric to determine the complexity of a language which is consistent with proven results.

For secured data transmission through the network, one of the methods followed is cryptography where encryption and decryption is carried out on text data. For cryptography key used is either symmetric or asymmetric. First parameter considered is the algorithm strength. Second parameter is the length of the key. Larger the size of the key, the security of the data is greater and lower is the data rate. The third parameter considered is the language complexity [4] by considering Telugu as the language. As the language becomes more complex, security increases, maintaining all parameters such as encryption algorithms and the key length constant. A comparative study has also been carried out with Bengali as a case study over Telugu and English. [7] Results showed that the retrieval percentage of text in Bengali is far less than English and Telugu.

In this work, a fourth security parameter have been added in the form of a dictionary file and a coded file [5]. This work has been carried out on eight Indian languages. Greater is the security, lesser is the percentage retrieval indicating greater is the language complexity. Hence security and percentage retrieval of various languages maintaining all factors constant such as encryption algorithm and key length can be used as a metric to determine the complexity of a language.

II. PROCEDURE FOR PAPER SUBMISSION

Vast studies have been carried out which helps cryptanalysts on breaking the cipher without knowing the key. Various languages across the globe consists of characters/symbols portraying different properties and behavior [3, 4]. Frequencies of occurrence of characters of a particular language are different. This helps cryptanalysts in breaking the cipher text with the help of frequency analysis by mapping of frequency of occurrence of each characters/symbol of cipher text to plain text. In the present work substitution cipher has been used. Here all plaintext letters will have one equivalent cipher text. The frequencies of cipher and plaintext frequencies will not be same, but the frequency count as a whole will be same. Thomas Jakobsen [2] has proposed a technique for fast cryptanalysis using substitution ciphers. Statistical analysis of the frequencies of bigrams, trigrams etc compared to monograms are found to be more useful in retrieving a portion of the plain text message. The cryptanalytic technique of enhanced frequency analysis has been developed [3] by combining the techniques of monogram frequencies, keyword rules and dictionary

Revised Manuscript Received on July 09, 2019.
Dr Devasish Pal, [1]Dept.of IT, MJCET, Hyderabad, India
Dr N V Ganapathi Raju, [2]Dept.of CSE, GRIET, Hyderabad, India
Mr Gautam Pal, *Security Lead, Intelematics, Melbourne, Australia*

checking. A sample plain text is encrypted using the proposed algorithm resulting in cipher text. Mapping of frequencies of all letters in the cipher text with that of the frequencies of letters in plain text are carried out. Next the replacement of cipher text letters is carried out with the mapped letters of plain text. The calculation of correct percentage retrieval with respect to plaintext has been indicated by K.W. Leeet.al [3].

In the present paper, eight Indian languages have been tested. They are Kannada, Malayalam, Tamil, Telugu, Gujarati, Bengali, Hindi and Punjabi apart from English language. For processing the corpus an indigenous software tool has been developed using Python scripting Language useful for all Indian languages. Python 2.7 has been used.

III. METHODOLOGY AND ARCHITECTURE

As seen from Fig 1, a preprocessed sample text of a fixed corpus size is taken. It is converted into a Dictionary file as explained in fig 2. The Dictionary file is encrypted using a 32-bit key using substitution method. Reverse Frequency mapping of occurrence of characters or code points of plaintext and cipher text dictionary file is carried out. Next percentage retrieval is calculated. The words of sample text are replaced by the extended ASCII values from dictionary file, which forms the coded file.

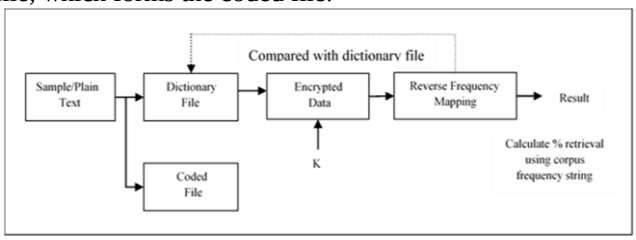

Fig 1: Dictionary Method Percentage Retrieval Model (DMPR)

. As seen from Fig 2, a preprocessed sample text of a fixed corpus size is arranged in decreasing order of occurrence. All the unique characters of corpus text considered constitute a corpus frequency string. Extended ASCII values from 33 to 252 are allotted to words arranged in decreasing order of occurrence which forms the dictionary file.

Processing of corpus: Pre-Processing of the corpus data has been carried out by eliminating white spaces, numbers, punctuation marks and special symbols. Single white spaces have been inserted in place of tabs. Removed all Non-language text characters. From the sample text considered the corpora for eight Indian languages has been split into a stream of words. Next, they are split into group of characters based on their Canonical Structure.

Corpus frequency string: All the unique characters of corpus plain text considered constitute a corpus frequency string.

Frequency analysis: It is the mapping of frequency of occurrence of each characters/symbol of cipher text to plain text. Every character displays a different frequency of occurrence and this value changes from language to language e.g. Frequency of occurrence of characters of English language is displayed in fig 4.

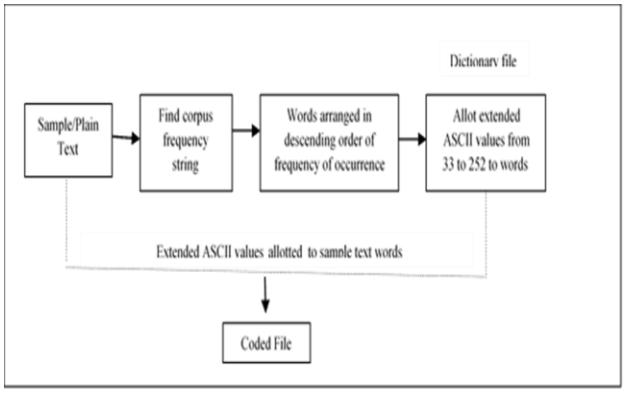

Fig 2: Dictionary file and coded file

The percentage is calculated for the occurrence of each different character of the plain text. The percentage of occurrence of every character of corpus frequency string is recorded. The individual letters of any language occur with greatly varying frequencies [2]. This helps cryptanalysts in breaking the cipher text with the help of frequency analysis by mapping of frequency of occurrence of each characters/symbol of cipher text to plain text. In the present work substitution cipher has been used. Thomas Jakobsen [2] has proposed a technique for fast cryptanalysis using substitution ciphers which uses the knowledge of diagram distribution of the cipher text. For security of text information through network, language complexity as a parameter was considered by Dr. Bhadri Raju MSVS [4] with a study on Telugu language. Sending information in Telugu language compared to English language was found to be more secure [4]. This concept was extended to eight Indian languages adding a fourth security parameter in the form of dictionary method [5].

Reverse frequency mapping: Here mapping based on frequencies of occurrence between the characters of plain text and cipher text is noted. This helps in the calculation of percentage retrieval of characters by cryptanalyst.

Cipher text: The plain text dictionary file is encrypted using an encryption algorithm using a 32-bit key of fixed length. Substitution cipher is used. To maintain consistency, same encryption algorithm and key has been used for the entire work.

Cipher frequency string: All the unique characters of cipher text considered constitute a cipher frequency string.

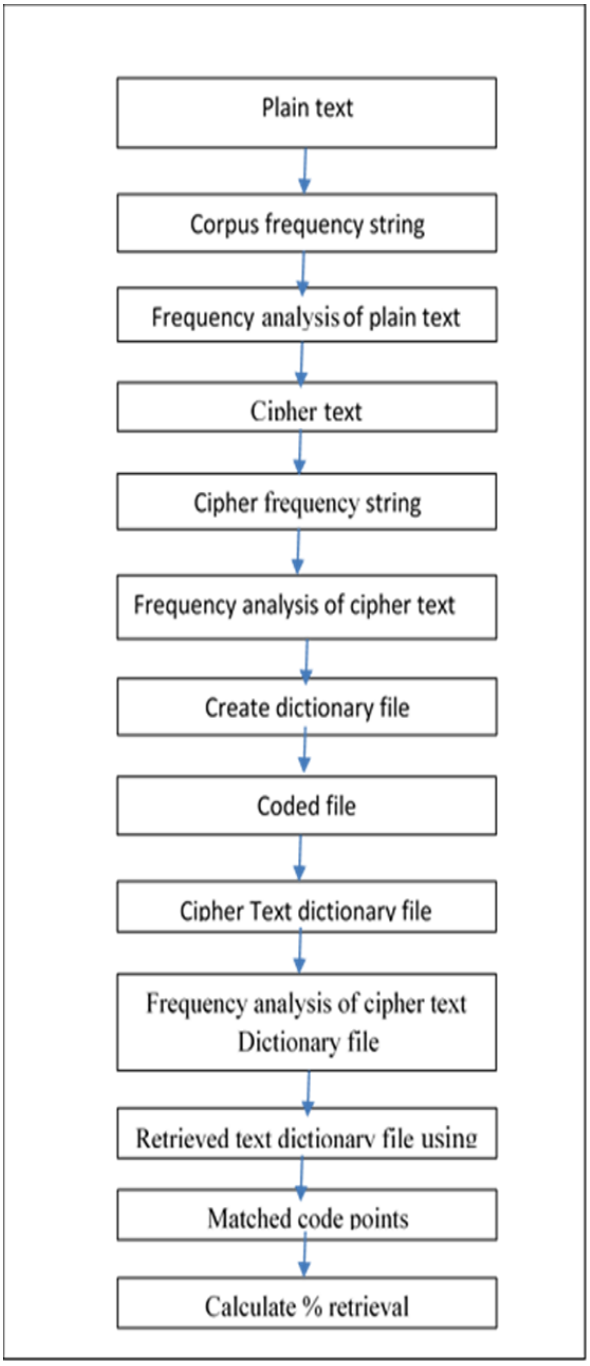

Fig 3: Architecture diagram of DMPR

Frequency analysis of cipher text Dictionary file: Frequency of occurrence of all characters or code points is noted.

Retrieved text dictionary file: Based on reverse frequency mapping between frequency analysis of cipher text and plaintext dictionary file, the cryptanalyst retrieves text dictionary file.

Matched code points: Comparing plaintext dictionary file and retrieved dictionary file, code points retrieved by cryptanalyst are noted.

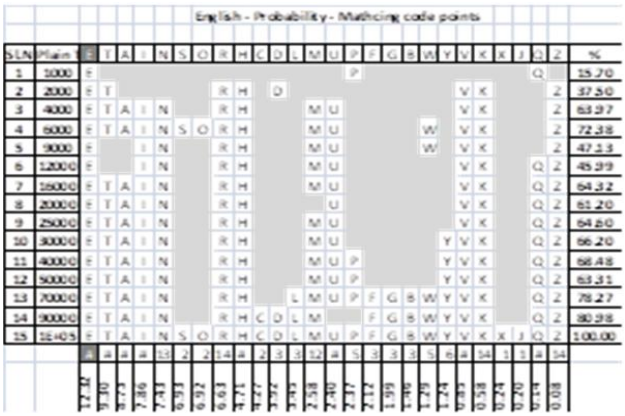

Fig 4. English Probability matching code points

Calculate % retrieval: Characters retrieved using frequency analysis and reverse frequency mapping are noted along with their occurrence percentages. Adding those percentages indicates the total retrieval percentage. For e.g. if the characters retrieved are: s, h, c and u. As per fig 1 the percentage occurrences of these characters are 6.93, 4.71, 4.21 and 2.40. The total retrieval percentage is 6.93 + 4.71+ 4.21 +2.40=18.25

Total length of corpus : 954
Retrieved plain text % is = 0.0450856627592

Sample *text:*

For communication through computer network, previously only English language using ASCII mode was used. Subsequently when Unicode was introduced, computer communication became a possibility for all language texts. This aspect generated interest in the field of language processing. Various studies have been carried out on language processing and its complexity issues. Various metrics were used to determine language complexity such as lexical density, morphological density, semantics etc. but there was no consistency in results. A language which appears most complex using one metric does not appear the same using other metric. This paper introduces a new metric to determine the complexity of a language which is consistent and with proven results. It introduces the concept of network security where using dictionary method, the percentage retrieval of an encrypted text using an encryption algorithm, fixed length of the key, fixed corpus size is calculated. Lesser is the percentage retrieval, greater is the security and language complexity.

Words arranged in decreasing order of occurrence frequency:

language ->8, the ->7, using -> 5, of ->5, is -> 4, was -> 3, and ->3, a ->3, complexity ->3, to ->2, density, ->2, security -> 2. fixed ->2, computer ->2, Various ->2, results. -> 2This ->2, introduces ->2, communication ->2, determine ->2, percentage ->2, metric ->2, an ->2, in ->2, which ->2, all -> 1, concept -> 1, encrypted ->1, paper ->1, through ->1, aspect -> 1, its -> 1, only -> 1, other -> 1, text ->1, network ->1, greater ->1, morphological ->1, not ->1, issues. ->1, mode ->1, texts. -> 1, ASCII -> 1consistent -> 1out -> 1appear -> 1for -> 1lexical -> 1does -> 1new -> 1dictionary -> 1, processing -> 1possibility -> 1English -> 1appears -> 1 for -> 1on -> 1carried -> 1where -> 1, length -> 1, became -> 1, Unicode -> 1, retrieval, ->

1, studies -> 1, previously -> 1 , one -> 1,size ->1, encryption ->1, there ->1, been ->1, introduced, ->1, interest ->1, used. ->1, but -> 1,Subsequently ->1, with ->1, corpus ->1, semantics ->1, were ->1, etc. ->1, calculated. ->1,key, ->1, metric. ->1, proven ->1, as -> 1 ,have -> 1 , no -> 1, algorithm, -> 1 , when -> 1 , same -> 1 , field -> 1 , complex -> 1 , Lesser -> 1 , A -> 1 , used -> 1 , complexity. ->1, metrics ->1, most -> 1, generated ->1, such ->1, network ->1 ,processing. ->1, method, the ->1, It ->1, consistency -> 1, retrieval -> 1

Dictionary File:

language -> !. the ->" ,using -> #, of -> $, is -> %, was ->&, and -> ', a -> (, complexity ->), to -> *, density, -> +, security -> ,, fixed -> -, computer -> ., Various -> /, results. -> 0, This -> 1, introduces -> 2, communication -> 3, determine -> 4, percentage -> 5, metric -> 6, an -> 7, in -> 8, which -> 9, all -> :, concept -> ;, encrypted -><, paper -> =, through ->>, aspect -> ?, its -> @, only -> A, other -> B, text -> C, network, -> D, greater -> E, morphological -> F, not -> G, issues. -> H, mode -> I, texts. -> J, ASCII -> K, consistent -> L, out -> M, appear -> N, for -> O, lexical -> P, does -> Q, new -> R, dictionary -> S, processing -> T, possibility -> U, English -> V, appears -> W, ï»¿For -> X, on -> Y, carried -> Z, where -> [, length -> \, became ->], Unicode -> ^, retrieval, -> _, studies -> `, previously -> a, one -> b, size -> c, encryption -> d, there -> e, been -> f, introduced, -> g, interest -> h, used. ->I, but -> j, Subsequently -> k, with -> l, corpus -> m, semantics -> n, were -> o, etc. -> p, calculated. ->q, key, -> r, metric. ->s, proven -> t, as -> u, have -> v, no -> w. algorithm, -> x, when -> y, same -> z, field -> {, complex -> |, Lesser -> }, A -> ~, used -> □, complexity. -> €, metrics ->ƭ, most -> , generated -> *f*, such -> „, network -> …, processing. -> †, method, the -> ‡, It -> ˆ, consistency -> ‰, retrieval -> Š,

Coded file:

X 3 > . D a AV ! # K I &i k y ^ &g .3] (U O : ! J 1 ?*f* h 8 " { $! † / ` v f Z M Y ! T ' @) H / ƭ o □ * 4 !) „ u P + F + n p j e & w ‰ 8 0 ~ ! 9 W , | # b 6 Q G N " z # B s 1 = 2 (R 6 * 4 ") $ (! 9 % L ' l t 0 ˆ 2 " ; $ … , [# S ‡ 5 Š $ 7 < C # 7 d x - \ $ " r - m c % q } % " 5 _ E % " , ' ! €

Language	% retrieval dictionary method		
	Corpus size		
	1000	5000	7500
English	0	16.54	15.86
Malayalam	0	12.04	11.77
Kannada	11.92	11.44	11.04
Telugu	5.62	10.79	11.02
Tamil	0	6.47	6.68
Bengali	6.06	5.87	0.08
Gujarati	0.16	0.13	0.06
Hindi	0.2	0.02	0.02
Punjabi	0.03	0.023	0.046

Table 1: % retrieval using dictionary method

Corpus	Word Tokens	Word Types	Type Token Ratio
Telugu	2,769,787	534,629	5.19
Kannada	3,118,988	474,067	6.58
Tamil	3,124,446	445,362	7.02
Malayalam	2313854	542,656	15.59
Marathi	1784197	196917	9.05
Urdu	117241	7782	15.05
Oriya	2966416	192463	15.42
Bengali	2531294	162,453	15.59
Punjabi	2308031	104367	22.12
Hindi	3,104,667	120,228	25.83

Table 2: Type Token Ratio

Language	Corpus size 1000	Corpus size 5000	Corpus size 7500	Corpus size 10000	Morphological Complexity	
English	0	16.54	15.86	15.61		
Malayalam	0	12.04	11.77	11.52	0.0643	
Kannada	11.92	11.44	11.04	11.12	0.151	
Telugu	5.62	10.79	11.02	11.45	0.194	
Tamil	0.0	6.47	6.68	6.26	0.142	
Bengali	6.06	5.87	0.08	0.06	0.065	
Gujarati	0.16	0.13	0.06	0.53		
Hindi	0.2	0.02	0.02	4.453		
Punjabi	0.03	0.023	0.046	0.079	0.0452	

Table 3: Linking of percentage retrieval using dictionary method with results of Central University, Hyderabad, India

Analyzing Table 3, the following is observed:

- A direct relation in regard to complexity of a language is observed with percentage retrieval of encrypted text.
- English language is the least complex and hence can be used as a reference to determine the relative complexities of various languages.
- The North Indian languages, Punjabi, Hindi and Bengali, where the percentage retrieval by a cryptanalyst is very low indicates that it is much more difficult for the cryptanalyst to retrieve data and hence more complex, maintaining all other parameters like encryption algorithm, length of the key, corpus size, language etc., Similarly the corresponding values under Morphological complexity and Lexical density with respect to Punjabi and Bengali are low compared to other languages.
- It is also observed that percentage retrieval using dictionary method considering the four corpus sizes namely 1000, 5000, 7500 & 10000 clearly indicates high values for Malayalam, Kannada, Telugu and Tamil with values declining in that order. Higher values indicate that the language is less complex. This pattern is observed in the case of independent work observations for the

same languages mentioned above with respect to Morphological complexity and Lexical density [6].

IV. CONCLUSION

Since a direct relation has been established between complexity of a language and percentage retrieval of an encrypted text, DMPR can be used as a metric to determine the complexity of a language with English language as reference as it is the least complex.

In future there is scope using the same Unicode for various languages, implementation with Predictable Partial Matching (PPM), a statistical technique using compression and decompression mechanism for better security of data and making the same metric more accurate to determine language complexity.

REFERENCES

1. Bao-Chyuan Guan, Ray-I Chang, Yung Chung Wei, Chia Ling Hu, Yu-Lin Chiu, "An encryption scheme for large Chinese texts", IEEE 37th Annual2003 International Carnahan Conference on Security Technology, pp.564-568, 2003.
2. Jakobsen, T, "A fast Method for Cryptanalysis of Substitution Ciphers", Journal of Cryptologia, Volume.19, Issue.3, pp.265-274, 1995.
3. Lee K.W., C.E. Teh, Y.L. Ta, "Decrypting English Text Using Enhanced Frequency Analysis", National Seminar on Science, Technology and Social Sciences, pp. 1-7, 2006.
4. Bhadri Raju MSVS, Vishnu Vardhan B, Naidu G A, Pratap Reddy L & Vinaya Babu A, "A Novel Security Model for Indic Scripts - A Case Study on Telugu", International Journal of Computer Science and Security, Volume.3, Issue.4, pp.303- 334, 2009.
5. Devasish Pal, Padiga Raghavendra, Dr. A Vinaya Babu: An Intelligent method of secure text data transmission through internet and its comparison using complexity of various Indian Languages in relation to data security – GJCST Vol 13 Issue 4 version 1 ISSN:0975 – 4172
6. G. Uma Maheshwar Rao, Christopher M and Parameshwari K: Introducing a measure of Morphological complexity in Indian Languages from the perspective of Morphological Analyzers - Center for Applied Linguistics and Translation Studies, University of Hyderabad.
7. Devasish Pal, Raju Ejjagiri, Dr. A Vinaya Babu: Complexity of Bengali Language and its relation to data security volume1 issue4-2012 (IJACIT) ISSN 2277-9140.

AUTHORS PROFILE

Dr Devasish Pal: After finishing BE(ECE) from College of Engineering, Osmania University, Hyderabad, served the Indian Air Force as Aeronautical Engineer (Electronics) for twenty years before taking premature retirement as Wing Commander. Worked in Defense labs (DRDO) as a consultant scientist for two years. Since then in the teaching line for the past 17 years. During the teaching profession completed M Tech (CSE) and PhD(CSE) in network security. Published a number of papers in International journals and presented papers in conferences. Presently working as a Professor in Computer Science in Muffakham Jah College of Engineering & Technology, Hyderabad.

N. V. Ganapathi Raju is working as a Professor in I.T. department, and Associate Dean Alumni Affairs at GRIET, Hyderabad. He received Ph.D from JNTUK, Kakinada. He received M.Tech (C.S.T.) from Andhra University. He has total 18 years teaching and 8 years research experience. His research interests include Information Retrieval and Natural Language Processing, Data Science and Machine Learning. He got UGC minor project grant MRP-4590/14 (SERO/UGC) in March 2014. He published research articles in various International journals. He is Webmaster for GRIET Website. Incharge for TASK, Spoken Tutorials-IITBombay, TEP admin for ISB for GRIET. He started Oracle Academy, CISCO Academy, Red hat Academy, DELL data science academy in GRIET for uplifting skill and knowledge of students.

Mr Gautam Pal: After completing BE(CSE) from Muffakham Jah College of Engg & Technology, Hyderabad worked in Infosys for six years and Sum Total for One year. Shifted to Australia on PR. Worked as Application security lead, designed and lead Telco and Fin tech teams into enterprise Agile DevSecOps practices with lower time to market and lower security risk. Over a decade of overall IT experience working on projects of top-rated companies like Macquarie Bank, Cisco, Telstra. Most recent experience has been leading a DevSecOps team Application Security to massive enterprise scale securely with 'guardrails not gates' cutting time to market.