International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 6123-6126
© Research India Publications. http://www.ripublication.com

An Enhanced CIA tree Using String Matching Algorithm

Renu S.[1] **and Dr. S. H. Krishna Veni**[2]

[1]*Research Scholar, Department of Computer Science, Noorul Islam University, Kumaracoil, Thuckley, Tamil Nadu, India.*

[2]*Assistant Professor, Department of Information Technology, Noorul Islam University, Kumaracoil, Thuckley, Tamil Nadu, India.*

0000-0002-8234-9980

Abstract

Data protection rules are varying in all over the world. Organizational rules and policies are the base for classifying unorganized files. Protection of data within and outside of the organization is a huge risk. Inorder to overcome this dicey situation organizations use different data protection techniques and policies. Organize data in a planned and productive way will helps to implement privacy effectively. We have proposed an enhanced CIA tree classifier which employs dictionary based pattern/string matching algorithms.CIA tree is a binary tree in which Confidentiality, Integrity and Availability are the three major level of the tree. Dictionary based string or pattern matching algorithms are the best solutions for automatic classification. We have proposed an effective string matching algorithm called StringCount where string counts are used for comparison. String count can be calculated by adding ascii equivalent of the alternate alphabets of the string.

Keywords: CIA tree,String Count, data dictionary, data classifier, sensitive data, protected data

INTRODUCTION

Today's information and communication industry uses number of data management methods and techniques, but studies and analysis shows that they are not up to the level. The frequently used methods in data classification are rule-based methods, probabilistic methods, SVM methods, instance – based methods and neural networks.

Data classification algorithms can be used to classify organizational data into sensitive, private, protected and public [1]. Some of the applications of string matching algorithms are ms word spell checker[2], matching DNA sequence, bioinformatics[3], Database queries, two dimensional mesh[4], language syntax checker, spam filter , plagiarism Detection[5] etc. Dictionary Based String Matching algorithms are used in virus scanning and network intrusion detection system. Ideal DBSM systems have high memory efficiency, high throughput and guaranteed performance. We can use various string matching algorithms for data classification.

RELATED WORKS

Sandeep K Sood et al[6] proposed a Sensitive Rating algorithm to classify data into sensitive, private, protected and public. The CIA factors were used for sensitive rating.

Renu S et.al [7] proposed a binary tree approach for data classification. A CIA tree with eight different paths was used to represent different security levels. The three different tree approaches are Simple binary tree approach, weighted tree approach and a complex network approach. A brain storming method is used to identify the security path. The simple tree approach is more accurate and time consuming compared to other two approaches.

Renu S et.al [8] proposed an automated data classification system for classifying organizational data. File sampling, key sampling, sample comparison and machine learning techniques etc were used to classify unorganized data. Data sampling, file sampling, file key comparisons and machine learning etc were the main techniques used in automated data classification system. The data classifier is a complex network which split files into small blocks. Collect file samples into file blocks and compare it with FK list using machine learning technique. Data training method were the core for classification.

Parikshit et.al. proposed [9] a model for 3 Dimensional Security in Cloud Computing. Confidentiality, integrity and availability are the three dimensions for data storage. This model offers three rings of security based on the sensitivity of data. An algorithm was used to categorize the data. The inner ring offer higher security and the outer ring have lower security. The middle offers intermediate security.

Rajagopal et.al proposed a model [10], at present security and business continuity are the crucial factors in the case of finance like levels.

Data technical insurance is a considerable factor during classification. Technical and background mining techniques can have vital roles along with cloud security techniques.[11][12]

Confidentiality, integrity and availability are the factors ensure security through indexing [13]. Through this we can optimize spatial factors [14][15].Proportional analysis on different technologies makes a system perfect[16].

PROPOSED WORK

AS the CIA tree [7] uses manual calculations, it is difficult to work in this technological world. Inorder to overcome this complexity, we have proposed an enhanced CIA tree classifier which uses dictionary based algorithm to classify unorganized data. The proposed model categorizes files into Sensitive, Protected, Private and Public based on organizational laws and policies.

It works like a machine learning algorithm where all the related patterns or string are stored in advance. The organizations can identify the files which they consider as critical or more sensitive. We can call these patterns as keys. The dictionary is a collection of keys in which the organization itself recognized it as the most valuable elements of their existence.

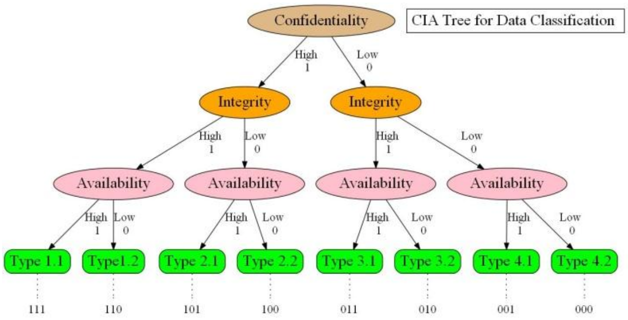


Figure 1: CIA tree for file classification.

Figure 1 shows a CIA tree with three levels where each level represents basic security factors such as Confidentiality, Integrity and Availability. Each node on the left sub-tree has well packed key dictionary and right sub-tree has less packed key dictionary. As the size of the dictionary increases; time, space and complexity also increase. Here, we are mainly focusing sensitive and protected data which are reside at the left side of the CIA tree. Four paths with path values 111,110,101 and 100 represent sensitive and protected data.

String Count Method

String or pattern matching algorithms are the best solution for data classification. We have proposed an enhanced string matching algorithm which uses string count methods for string comparison.

The string count pattern matching algorithm uses ascii equivalents for counting the strings or patterns.

Algorithm

String Searching

```
1.  Read main string , S
2.  Read pattern, P
3.  Calculate length of the strings ,
    Let t1=S.length ();
    Let t2 = P.length();
4.   If(t1== t2)
        a.  For i=0 to i<t1
        b.  Read characters of the string S, c
        c.  Read characters of the string P, c1
        d.  Change character c to ascii, j
        e.  Change character c1 to ascii, j1
        f.  If((i+2)%2 == 0)
                i.  k=k+j
               ii.  k1=k1+j1
                    End if
                 Else
                       i.  l=l+j
                      ii.  l1=l1+j1
                           End if
            End if
        g.  Else if(t1<t2)
                1. Calculate the index of substring,
                   index = S.indexOf(P)
               2. If (index <0)
                      a. Print"Substring not found"
                       b. Exit
                        Else
                       a.  Print "Substring found in
                           index"; index
                  Exit
                  End if

                    End if
5.  If(k==k1)
        a.  If(l==l1)
                i.  Count++;
             End if
     End if
6.  If(count ==1)
        a.  Print "Match Found"
      Else
        b.  Print "No match Found"
7.  Stop
```

S and P represent main string and pattern string respectively. Even and odd position alphabets are count separately by using its ascii equivalent .If a match occurs at the first count check with the second count. Matching both the counts means that the strings are same.

CIA classification needed dictionary based string-count method to accomplish automated categorization. There is a minute change when we use a dictionary as a reference. The comparison process is mentioned below:

```
Procedure StringComparison()
1. While (K1[x] != l && x <dictionarySize)
        a.X++
   end while
2. if(k==l1[x])
        Count++
   Enf if
End Procedure
```

Dictionary creation

A pattern dictionary is needed to accomplish the automated characteristics of CIA tree. The dictionary contains related patterns with string count. The related patterns help to discriminate files as sensitive and protected. An altered string counting algorithm is used to create a pattern dictionary and string processing. Dictionary creation and storing keys to the dictionaries are risky. A little fault or mistake makes our system completely worse. Storing keys in the dictionary seems highly responsible and it should be accurate. As the size of the dictionary increases, accuracy also increases. String comparisons are performed inside each node of the CIA tree. String comparison is simple and quick.

Dictionary can be sorted or unsorted. Creation of sorted dictionary is hard compared to unsorted dictionaries, but the operation will be ease and quick by using binary search method. In the case of unsorted dictionary, creation is sudden and simple but linear search is applicable in string processing which is time consuming.

PERFORMANCE ANALYSIS

Performance analysis of StringCount algorithm and automated data classifier is discussed in this section. If the string length of the main string and pattern string are same, we can use the StringCount method. Even and odd position characters of the strings are processed separately.

Main String, S: AABB

Pattern String, P: AABC

String lengths of both the string are same. So we can use string count method. The upper case letters are changed into lower case.

S : aabb

P: aacb

Convert into ASCII form

S : 97 97 98 98

P: 97 97 98 99

K=0,K$_1$=0, L=0 , L$_1$ = 0

After StringCount the value of K, K1,L and L1 will be

K= 195

K1 = 195

L = 195

L1 = 196

Here. K==K1 i.e., 195 == 195

But L!= L1 i.e., 195 != 196

The strings are different.

The algorithm is suited for string having same length.

Library functions are used for substring searching. The StringCount algorithm is apt for dictionary based string matching because single comparison is performed if there is no match found and double comparison is performed if there is a match occurs.

The time complexity of the StringCount algorithm is O(m) where m is the length of the main string. In dictionary based string matching algorithm pattern string is already processed and StringCounts are stored in advance. The space complexity of average and worst case is O (n) where n is the size of the dictionary. The time complexity of average case is O (log n) and worst case is O (n).

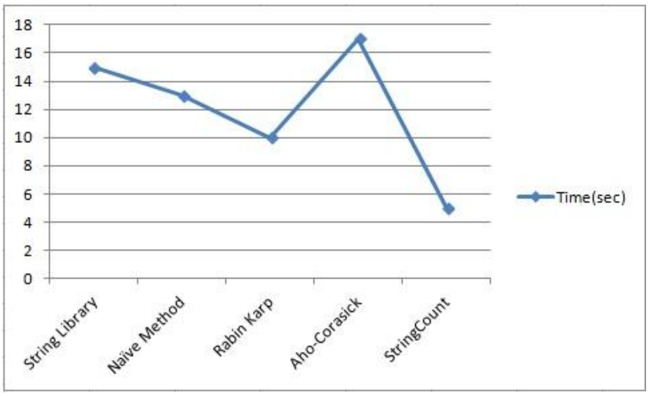
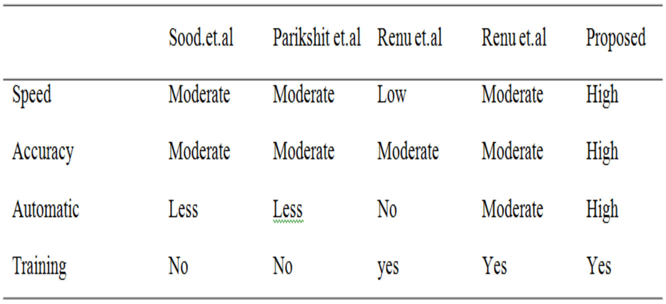


Figure 2: Performance graph of string matching algorithms.

We made a comparison with the existing methods such as string library method, Naïve method, Rabin Karp and Aho-Corasick. The analysis graphs shows that StringCount method takes less time compared with the existing string classification methods.

Each security methods use data classification as a prior step.

We have analyzed the enhanced CIA tree with some of the most efficient methods which uses classification at the early stage and observations are explicit at the fig.3.



	Sood.et.al	Parikshit et.al	Renu et.al	Renu et.al	Proposed
Speed	Moderate	Moderate	Low	Moderate	High
Accuracy	Moderate	Moderate	Moderate	Moderate	High
Automatic	Less	Less	No	Moderate	High
Training	No	No	yes	Yes	Yes

Figure 3: Performance analysis

Compared to the existing data classification systems, performance of the proposed system is excellent. An automatic system using dictionary approach can produce excellent output. Speed and accuracy of the existing system is average compared with the new one.

CONCLUSION

A dictionary based string matching algorithm using StringCount method can produce accurate output with less time. Odd and even position values of the string can be calculated using its ascii equivalents. Dictionary is a collection of keys which represent critical data in the organization. This method is suitable for comparing string with same length.

REFERENCES

[1] The California State University (CSU) 8065.S02 Information Security Data Classification.

[2] Alberto Apostolico and ZviGalil," *Pattern Matching Algorithms*" Published in Oxford University Press, USA, 1st edition, May 29, 1997.

[3] Lok-Lam Cheng, David W. Cheung and Siu-Ming Yiu," *Approximate String Matching in DNA Sequences*", In Proceedings of the Eighth International Conference on Database Systems for Advanced Applications (DASFAA'03), pp. 303-310, 26-28 March 2003.

[4] Nimisha Singla,Deepak Garg"String Matching Algorithms and their Applicability in various Applications", International Journal of Soft computing and Engineering,Volume1,issue-6,January 2012.

[5] Ramazan S. Aygün "*structural-to-syntactic matching similar documents*", Journal Knowledge and Information Systems, ACM Digital Library, Volume 16 Issue 3, pages 303-329, Aug 2008.

[6] SANDEEP K SOOD ,"A combined approach to ensure data security in cloud computing ",Journal of Network and Computer Applications 35 (2012) 1831–1838, 2012.

[7] Renu S,"A Novel Method to Classify Organizational Data Using CIA Tree approach", IEEE Explore, *27 nov 2015*, DOI :10.1109/GET.2015.7453838 , Pages 1-5.

[8] Renu S,Dr S H Krishnaveni "An Enhanced Automated Data Classification System Using Complex Network", IJCTA, 8(5), 2015, pp. 2301-2306

[9] Dimensional Security in Cloud Computing parikshit Prasad badarinath oja IEEE 2011.

[10] Rajagopal, N. " A new data classification methodology to enhance utility data security " ; Tata Consultancy Services Ltd., Mumbai, India ; Prasad, K.V. ; Shah, M. ; Rukstales, C. Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES, Pages: 1-5.

[11] Nguyen Truc Mai Anh , Thi Ngoc Chau "Towards a robust incomplete data handling approach to effective educational data classification in an academic credit system" IEEE international conference on Data Mining and Intelligent Computing (ICDMIC), 2014,pages:1-7

[12] Zardari, Tronoh,; Low Tang Jung ; Zakaria, "Hybrid Multi-cloud Data Security (HMCDS) Model and Data Classification" M.N.B. Advanced Computer Science Applications and Technologies (ACSAT), 2013 pages: 166-171.

[13] Fatemi Moghaddam, F. ; Selangor,; Yezdanpanah, M. ; Khodadadi, T. ; Ahmadi, M "VDCI: Variable data classification index to ensure data protection in cloud computing environments". Systems, Process and Control (ICSPC), 2014,pages:53-57.

[14] Yushi Chen ; , Harbin, Xing Zhao ; Xiuping Jia "spectral–spatial classification of hyperspectral data based on deep belief network" *IEEE Journal of Applied Earth Observations and Remote Sensing*, *Volume:8 , Issue: 6 ,2015,pages: 2381-2392.*

[15] Paiva, J.G.S. ; Uberlandia, Schwartz, W.R. ; Pedrini, H. ; Minghim, R. "An Approach to Supporting Incremental Visual Data Classification "IEEE Transactions on Visualization and Computer Graphics, Volume:21 , Issue: 1,2015, pages: 4-17.

[16] Kumathekar, C.N. ; Chavan, A.P. "Efficient Intrusion Detection System Using Stream Data Mining Classification Technique" Desale, K.S. ; IEEE conference on Computing Communication Control and Automation (ICCUBEA), 2015 pages:469-473.