

Intelligent Data Security - Efficient Way to Store the data

Prof.Sushopti Gawade
P.I.I.T. Panvel.
sushopti@gmail.com

Prof. Swati Chopade
V.J.T.I. Matunga.
swati.chopade@gmail.com

Mr. Aniket patil
S.C.O.E Kharghar
anike1989@gmail.com

ABSTRACT:

Data Security has always been the need of many business sectors. Especially in banking area's where the critical information of the customer's is there with the banks. Increasing Cyber-attacks, network spoofing, key logging, malwares, viruses are the Threats to data. In this paper author proposed an algorithm which will arrange the data in a particular format and will make it complex as well as space efficient. The main intention of this algorithm is to provide security as well as saving storage space. If more security is needed we can encrypt the data for guaranteed secure data. It can be stated that algorithm is suitable for making data Complex and tiny.

KEYWORDS : *Data Security,space Efficient,network spoofing.*

I. INTRODUCTION

Let us begin the topic with the basic tools which we use for storing our data. These tools includes notepad, WordPad, .Microsoft access, Microsoft Excel. If we consider the data storage in private as well as in public organizations then it turned out to be same as mentioned above. And if we look the biggest data storage then ERP, CRM, Cloud Computing, Web servers will come in to picture. Our concern is not to look at particular format but to make Data secure irrespective of its format. Data Security has always been the need of many business sectors. Especially in banking area's where the critical information of the customer's is there with the banks. Increasing Cyber-attacks, network spoofing, key logging, malwares, viruses are the threats to data. But to make our point we will consider a Scenario where we will be dealing with the data which is in Microsoft Excel (.xls) format or in (.csv) format. In this context I applied my algorithm in java converter using java language.

II. MATERIALS NEEDED

A. Microsoft access file with data in it

As every program needs data to be processed . But now a days people are worried about the security of data, as normal human being who is little bit literate with computer knows text is available in either wordpad or Microsoft word.sometimes it can be stored in table formats .Then in this case people can prefer excel sheets. Consider that we are dealing with the Bank Database which is in .xls format,means it is available in excel sheet format. So it is obvious that the excel sheet will consist of customer records including their bank details

account number, pan number as a very sensitive and important data, personal information (address) etc.

B. File converter designed in Java

As the name suggests , this is the tool which will be reading Excel files as a input source and it will convert those files in the text files. The converter baicsally reads file field by field and then it will keep on converting evryfield in text format.

C. Software requirements

We are working on banking information so to get input file we need Microsoft office with excel data sheet.Converter tool has been designed in java .To execute the above converter safely we will require windows xp or windows seven and java installed in it. And also Microsoft office pc suite is also need to be installed in pc to open the desired Excel file.

III . METHODS

Now let us begin with the actual conversion logic of our data. What we will do is that, we will be declaring an array which will be basically two dimensional. And with the help of the array we will be reading data line by line.

A. Reading data by Field(column)

The array will be designed in such a manner that it will be reading the data field by field that is we have to read single cell or field at a time.

B. Keeping the track of Field Number

In this case while reading the Strings which will be present in each field we will keep track of filed number.eg. customer's first name will be field number 1 in the bank case.

C. Assigning unique code for every field

Here we will assign unique code to every field.eg. Suppose if I have field number 6 as date then I'll assign "10311" as a unique code to date field.

D. Calculating the String length of each string

we will calculate the string length of each and every string which we read from array specified in previously.eg.06aniket here string aniket has 6 characters so term 06.

IV. ALGORITHM

Now we are at the most important step of our topic, the algorithm which is the heart of every project or every theory or concept. It's a proof of every proposed theory which consists of logic.ultimately it gives us the proper sequence of instructions to follow to achieve our goal.

Let us discuss our algorithm which will be step wise given as below,

Step 1: Pick the first field

Step 2 :Trim leading and trailing spaces if any

(Leading and trailing spaces are the spaces which are present before start of string and after end of the string, but not in between the string).

Step 3 : Check if the field is blank

Step 4 : If field is blank then skip to next field

Step 5 : If field is not blank then field tag

Step 6 : Calculate the String length

Step 7: If the length is less than 10 then add the field tag to it. And also append "0" to the new string in addition to that add field length and in the end add the main string.

Step 8 : If field length is not less than 10 then only add field tag, length of field and main field itself.

Step 9 : Keep on doing for each and every line.

Step 10 : After reading the whole file join the each and every field read with their conversions. E.g. After adding field tag+length+field join each and every field.

Step 11: Output will be in the form of mixed content numbers as well as letters.

Step 12: Now save this output in text file.

V. RESULTS

1. We read the excel file field by field using java language. The sample of file used is given below

Consumer Name 1	Consumer Name	Consumer Name 3	Consumer Name	Consumer Name	Date of Birt	Gender	PAN	Passport N	Voter I	Phone No #1
SHANKAR	IYER				17041981	2	AAPI0958F			28593660
MAHESH	RAJAN				8041978	2	AHPR0070Q			8025302133
MAHESH	RAJAN				8041978	2	AHPR0070Q			8025302133
KALUSHAL	KHURANA				9081982	2	APVK2344E			30889574
KALUSHAL	KHURANA				9081982	2	APVK2344E			30889574
HARISH	KUMAR	HANDA			8061970	2	AATH6552P			951204056100
HARISH	KUMAR	HANDA			8061970	2	AATH6552P			951204056100
PRAVIN	RAWOOL				30081981	2	AHPR6363A			400081326
PRAVIN	RAWOOL				30081981	2	AHPR6363A			400081326
SANDEEP	TIWARY				25111981	2	AFBPT9566J			1141828073
SANDEEP	TIWARY				25111981	2	AFBPT9566J			1141828073
VIMLA	GUPTA				26081980	2	AHJP68028M			43689700
VIMLA	GUPTA				26081980	2	AHJP68028M			43689700
ANKIT	GARG				27041981	2	AHJP2395J			32368672422416
ANKIT	GARG				27041981	2	AHJP2395J			32368672422416
RAKESH	SHARMA				19041980	2	FORM60			9850815324
RAKESH	SHARMA				19041980	2	FORM60			9850815324
YOGESH	LAXMANBHAI	PATEL			17111963	2	AFKPP9327A			22354453
YOGESH	LAXMANBHAI	PATEL			17111963	2	AFKPP9327A			22354453
SATYANARAYAN	BALASUBHARMAN				8101975	2	AFSP8859E			252826967258110
SATYANARAYAN	BALASUBHARMAN				8101975	2	AFSP8859E			252826967258110
DARSHANA	H	SHAH			10051983	1	BKCP5457F			255113366079460
DARSHANA	H	SHAH			10051983	1	BKCP5457F			255113366079460
SACHIN	PRAMOD	HATE			16101970	2	ABDQ1894TQ			22268409922684000
SACHIN	PRAMOD	HATE			16101970	2	ABDQ1894TQ			22268409922684000

2. Then we converted that file using our logic. And we copied that converted code into notepad which is as given below:

```

AXZK11BF02150001ABCDBANK
310120094CX5200000
PN03N010107SHANKAR0304IYER070817041981080
120910AAOPI0958F120828593660PA03A010113APN
INDIA LTD0225NO B-804 SAGAR TECH
PLAZA0318ANDHERI KURLA
ROAD0421SAKINAKA ANDHERI
EAST0506MUMBAI0602270706400072TL04T001011
0BF021500010208CITIBANK03165189369370841000
040210050110808240420080908050920081108310120
091205243381304-25015010260201ES02**

PN03N010106MAHESH0305RAJAN07080804197808
0120910AIHPR0070Q12108025302133PA03A010123
NO 51/2 MILITARY SCHOOL0216MUSEVA ROAD
POST0313RICHMOND TOWN0410NR
STADIUM0509BANGALORE0602290706560025TL0
4T0010110BF021500010208CITIBANK031651893693
708430000402100501108082104200811083101200912
0101301015010260201ES02**TRLR
    
```

International Journal of Computer Architecture and Mobility (ISSN 2319-9229) Volume 1-Issue 7, May 2013

VI. CONCLUSION

As discussed above this paper suggests the best way to convert normal excel file information into encrypted text file. We have considered simple banking customer information as a case study and tried our algorithm. Results are already shown in above V. Here basically we achieved data security by encrypting information and reduced the space required to store excel sheet information.

The Data is read from Excel file and is converted to encrypted text file. The idea behind copying the Data to text file is to save space which is previously used by large excel files (assuming the data present in those files are in large amount)

When we saved our encrypted Data in text file the file size decreased as it's a text file. So less Data storage space is required. And performing certain actions on data will result into encrypted complicated code format Data.

More over if we need extra security then we can also apply some encryption algorithm on such Data to make it guaranteed secure. Even if somebody managed to hack these files over a network access, he/she will not be able to decrypt the file. And data will be secure.

VII. FUTURE SCOPE

This logic (Algorithm) can be implemented over many languages. In some situations it will need some modifications but it will surely work. This can be modified to work better by using different encryption algorithms in distributed networks, which aimed to achieve more compression in data and ultimately saves space.

VIII. REFERENCES

A new algorithm is proposed which will give you data encryption as well as reduce data size.

- [1] http://www.csvreader.com/java_csv_samples.php
- [2] <http://www.java-examples.com/parse-csv-file-using-stringtokenizer-example>
- [3] <http://www.vogella.de/articles/JavaExcel/article.html>
- [4] <http://www.andykhan.com/jexcelapi/>
- [5] <http://poi.apache.org/>
- [6] <http://www.roseindia.net/tutorial/java/poi/readExcelFile.html>
- [7] <http://www.java.com/en/download/>
- [8] http://en.wikipedia.org/wiki/Java_%28programming_language%29
- [9] <http://download.oracle.com/javase/1.4.2/docs/guide/security/jce/JCERefGuide.html>
- [10] <http://www.jasypt.org/>