

DESIGN OF THE FILE SEPARATION STRUCTURE FOR EXCEL VBA PROGRAM

Seung Ju Jang

College of ICT Engineering, Dong-Eui University

Korea

sjjang@deu.ac.kr

ABSTRACT

We use a lot of Excel programs in our daily work. Then, VBA program is used for data processing in Excel. Excel VBA program works in conjunction with Excel function. The fact that VBA program and Excel work together is a disadvantage. This paper designs a function to protect VBA programs used in Excel files. I separate the VBA module from the excel file. We want to protect important programs by isolating VBA modules. To protect the VBA module in the Excel file, separate the VBA module from the Excel file. By doing this, unauthorized users are prevented from accessing the VBA module itself. Only the user who is allowed access to the VBA module is designed to be able to use the VBA module with normal use. Experimental results show that the proposed method works well.

Keywords: Excel VBA, VBA Program File Separation, VBA Module, Excel.

1. INTRODUCTION

This paper is related to Excel VBA which we use frequently in everyday life. Excel VBA(Visual Basic for Applications) is used to automate repetitive operations on data in complex Excel sheets. Excel VBA is basically built into Excel. VBA makes it very easy to do complex calculations. The Excel program has a built-in Visual Basic language inside and allows programmers to write their own program code that works inside the program.

Thanks to VBA, users can go beyond the limitations of the program itself and perform better tasks. VBA is closely tied to Visual Basic and uses the Visual Basic runtime library. However, VBA code can generally be run only within a host application, not from an independent executable program. Excel VBA uses OLE automation to allow one application to be controlled by another application.

VBA is often used in Excel spreadsheets for complex data processing. In this paper, I isolate VBA module in Excel to ensure more secure operation of VBA module.

Recent Excel spreadsheets have an OLE structure. OLE (Object Linking and Embedding) files (or COM Structured Storage) formats have been used as document formats for Word, Excel, and PowerPoint of the Microsoft Office. The OLE file format has the same file system structure as an internal small hard disk [1].

In this paper, I propose a method to protect the program by designing a separate structure from the Excel file for the module generated for the VBA program used for data processing in Excel.

I describe related works in chapter 2 of this paper, VBA program separation structure system in chapter 3, experiment in chapter 4, and conclusion in chapter 5.

2. RELATED RESESRCHES

The VBA program provides a convenient way to process data quickly in an Excel sheet. VBA is commonly used to process Excel data quickly. When you run a VBA program, the VBA code that is programmed in the Excel file is stored in vbaproject.bin. The internal structure of the vbaproject.bin file is as follows [2, 3, 4].

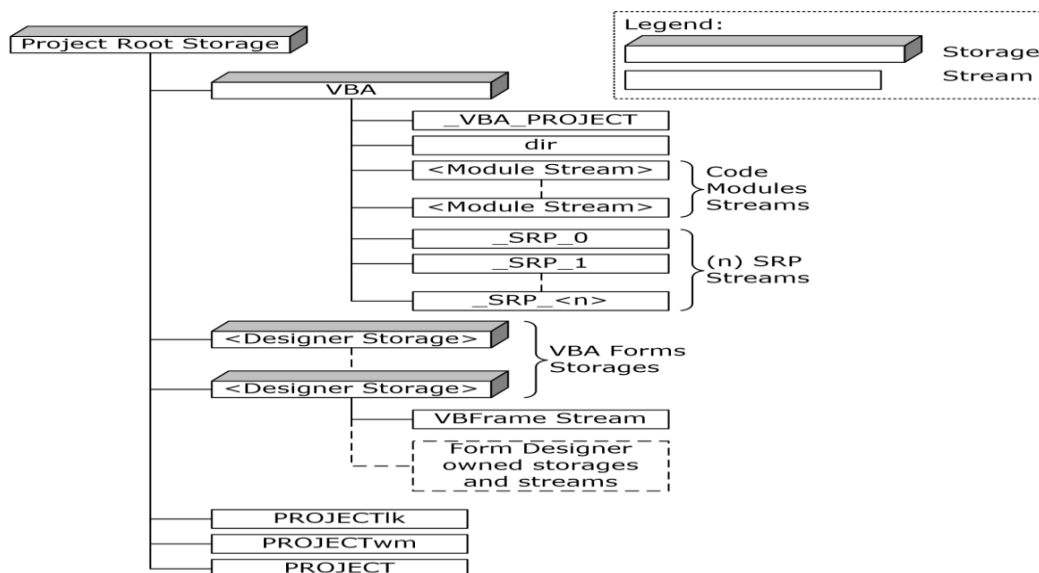


Figure 1 Vba Storage Hierarchy

As shown in Figure 1, the vba storage hierarchy specifies the VBA project and the project items it contains. All data is stored in the structured storage specified in [MS-CFB]. The repository and streams must be organized according to a hierarchy rooted in Project Root Storage.

<Project Root Storage>

This is root storage. It has a VBAStorage and a project stream. In addition, it can additionally have various streams.

<VBA Storage>

VBA storage defines VBA project and module information. It has a VBA_PROJECT stream and a dir stream.

<_VBA_PROJECT Stream>

It manages version information about project information. It must have specified data by the VBA_PROJECT stream.

<dir stream>

This specifies ownership of the VBA project, project references, and module ownership. It should have the specified data by the dir stream.

<Module stream>

This specifies the source code module within the VBA project. The name of this stream is specified by the MODULE STREAM NAME.

3. DESIGN of the FILE SEPARATION STRUCTURE of VBA PROGRAM

The use of VBA programs in Excel has a convenient aspect. The advantage is that you can program without having to install a special compiler. It also has the advantage of being able to easily program the desired functions when needed.

However, these convenient functions are also a fatal weakness in maintaining the security of the program source code. In particular, VBA programs created using Excel data often exist in the same file, so it is necessary to protect the source code.

I extract the vbaproject.bin file from the existing Excel file. The extracted vbaproject.bin file is separated from the existing Excel file. And, instead of a extracted vbaproject.bin file, I install a module that can display a message that a vbaproject.bin file is needed. When this message is displayed, the vbaproject.bin file is added to the Excel file. If the vbaproject.bin file is normally added to the existing Excel file, normal operation is possible.

I extract the vbaproject.bin file from the VBA programmed Excel file. The vbaproject.bin file extracted from the Excel file is managed separately. In addition, I load a guide module for use. Instead if the user wants to use this module.

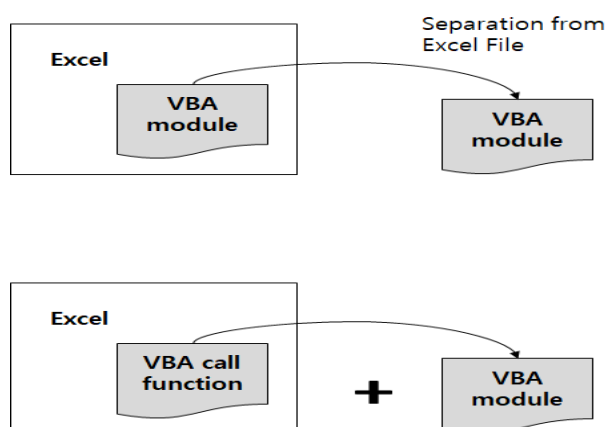


Figure 2 Vbaproject.bin File Separation Structure

It is designed to be run as shown in the Fig. 2 after the program is written in the VBA program module and the final writing is completed. By doing so, I can remove the VBA program module from an existing Excel file.

This separation allows the program source code to be safely stored and managed by managing only the VBA program module separately.

In order to be operated as above structure, I design the function to extract vbaproject.bin file in Excel file. I design the function of inserting the module indicating that there is no VBA module, when the vbaproject.bin file extraction is completed. When a user wants to use the corresponding function in Excel, he is informed that there is no VBA module. Then, I design a vbaproject.bin file insertion module that allows the user to use the vbaproject.bin file conveniently.

The design of the function to separate the vbaproject.bin file from the excel file is as Fig. 3. First of all, in the case of a VBA program in an Excel file, it is stored in a macro format.

Excel files saved in macro format are in compressed file format. Therefore, the process of decompressing the compressed file is performed.

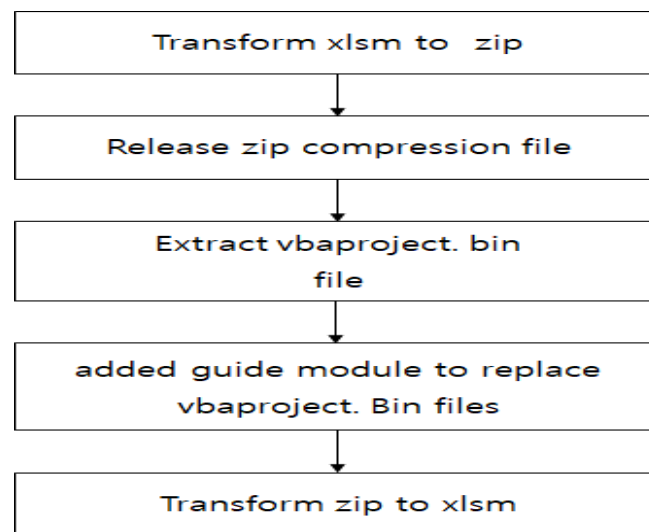


Figure 3 Separation Procedure vbaproject.bin from Excel File

I design a function to insert a module to indicate that there is no VBA module, once the vbaproject.bin file has been extracted, If a user does not have a VBA module in Excel, user can not process the data they wants. Thus, I design a module to guide this.

I design a function to add VBA module, if Excel does not have a VBA module. The function of adding VBA module to Excel is as Fig. 4.

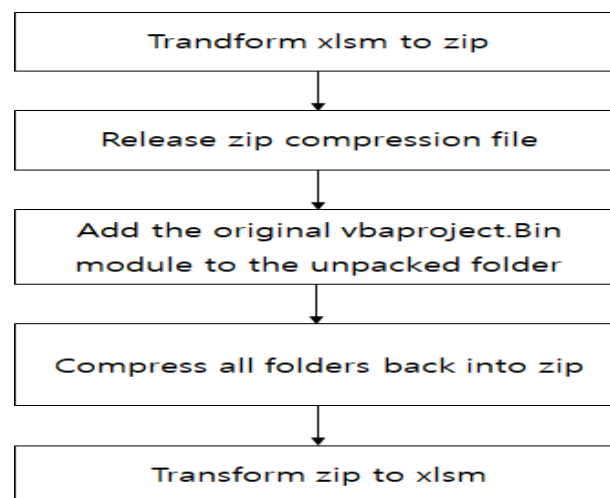


Figure 4 Adding vbaproject.bin to Excel File

The above Fig. 4 shows the process of adding VBA program module to Excel file. It changes the original Excel file to the zip extension. It unzips the changed zip file. If you unzip the file, there is an xl folder. you add the original vbaproject.bin file to this xl folder. When the vbaproject.bin file is added, you zip all folders and files. When zipped, you change the extension to xlsx.

First, in order to add the vbaproject.bin file in the excel file, you change the extension xlsx of the excel file to zip. Then you open the zip compressed file (open). You open the compressed file and add the vbaproject.bin file to the compressed file. You add the vbaproject.bin file and

save the zip file. After saving the zip file, you convert again the zip extension to the original extension xlsx.

4. EXPERIMENT

In this paper, I implement a program for module design that extracts vbaproject.bin file from excel file. The operation process of the implemented program is as shown in Fig 5 and Fig. 6.

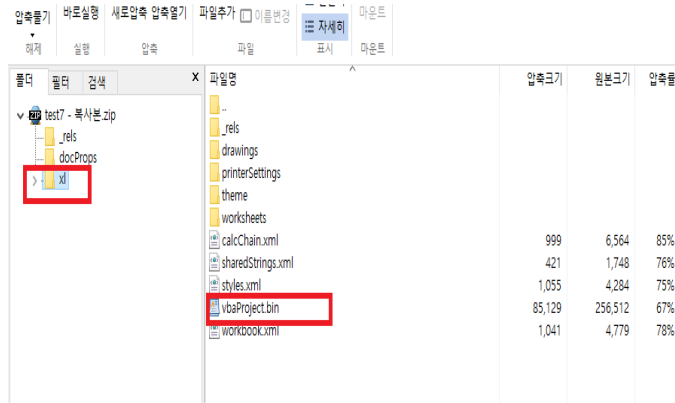


Figure 5 Extraction of vbaproject.bin File

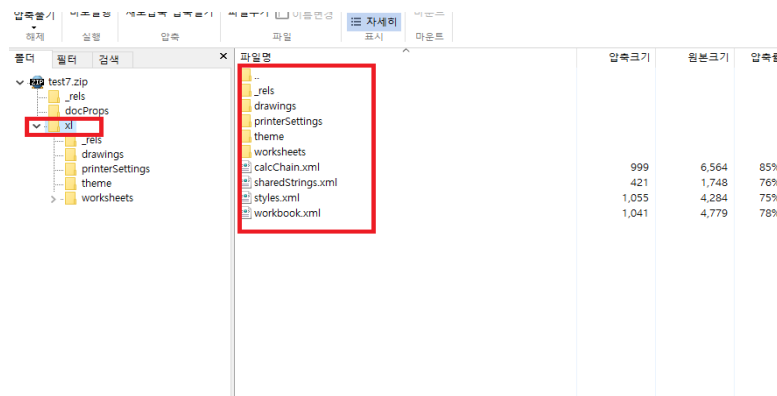


Figure 6 Contents of Extracted vbaproject.bin File

Fig. 5 and Fig. 6 is the process of extracting only the VBA program module, in the case that the VBA program is included in the Excel file. First, we uncompresses the compressed file format. Then we extracts only the vbaproject.bin file from the extracted file.

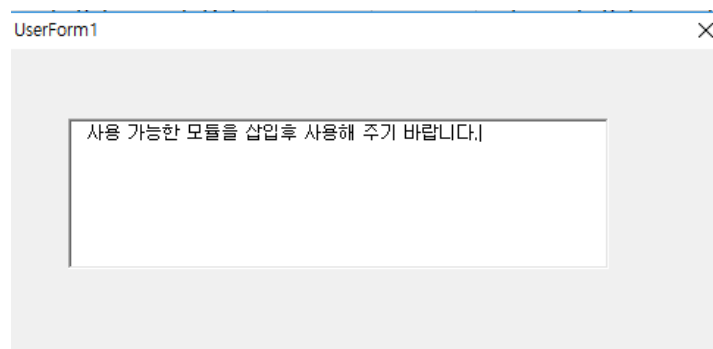


Figure 7 Screen to guide Module Insertion

The above Fig 7 shows that there is no VBA program module in the Excel file. By doing so, the user can insert a VBA program module to enable normal program operation.

If there is no VBA program module in the Excel file, normal data processing is impossible. You must add the VBA program module to the Excel file. After opening the compressed Excel file, you add the vbaproject.bin file to the excel file. The folders and saved files after the addition are shown in Fig 6.

We then return the compressed file format to the original Excel file format. Then, the VBA program module originally created in the Excel file will be able to operate normally. As a result of the above experiment, it was confirmed that the design of the separated structure for the VBA program used for data processing in Excel works normally.

5. CONCLUSION

In this paper, I try to protect important technology by separating VBA program file from Excel file in using Excel VBA. It is generally composed of Excel data files and VBA files that process data. It protects the VBA program module to the extent that it sets the password to protect it.

In this paper, I design a file separation structure for VBA program to protect VBA program module which is important.

We extract the vbaproject.bin file from the existing Excel file. The extracted vbaproject.bin file is separated from the existing Excel file. And, instead of a separate vbaproject.bin file, we install a module that can display a message that the vbaproject.bin file is needed as Fig. 7. When this message is displayed, the vbaproject.bin file is added to the Excel file. If the vbaproject.bin file is normally added to the existing Excel file, normal operation is possible. We extract the vbaproject.bin file from the VBA programmed Excel file. Then we load a module that can call the vbaproject.bin file instead of the vbaproject.bin file extracted from the Excel file.

In this paper, I implement a program for module design that extract vbaproject.bin file from excel file. As a result of the experiment, it was confirmed that the designed structure of the VBA program used for data processing in Excel works normally.

We will develop this research and implement the program so that it can be used conveniently for the modules designed in the paper.

REFERENCES

- [1] Jang, Seung Ju, "DESIGN OF THE EXCEL VBA PASSWORD CHANGING MODULE", European Journal of Engineering and Technology, Vol. 6 No. 2, 2018.
- [2] <http://www.reversenote.info/ole-file-format/>
- [3] Hong, Sug Myung, "Detection of Obfuscated VBA Macro in Microsoft Office Documents", Department of Computer and Radio Communications, Engineering, Korea University, Thesis for the Degree of Master, May 2017.
- [4] Suryam Tiwari, Vijay Shrivastav, "Development of Design Spreadsheet Tool for R.C.C. Beam Design using V.B.A., IJIRST –International Journal for Innovative Research in Science & Technology| Volume 4, Issue 11, April 2018

- [5] Michael R. Bartolacci, Larry J. LeBlanc, Yasanur Kayikci, Thomas A. Grossman, Optimization Modeling for Logistics: Options and Implementations, *Journal of Business Logistics*, Volume 33, Issue 2, June, 2012 pp.118–127, 13 June, 2012.
- [6] J Pichitlamken Email author, S Kajkamhaeng, P Uthayopas, R Kaewpuang, “High performance spreadsheet simulation on a desktop grid”, *Journal of Simulation*, Volume 5, Issue 4, pp266–278, November, 2011.
- [7] Zhiyong Zhang, “Construction of the Multimedia Teaching Platform of Cost Accounting Course Based on EXCEL VBA Program”, *International Journal of Emerging Technologies in Learning*, Vol. 13, No. 05, pp. 177-187, May, 2018.
- [8] R Eso, L O Safiuddin, L Agusu, L M R F Arfa, “Simulation of 2D Waves in Circular Membrane Using Excel Spreadsheet with Visual Basic for Teaching Activity”, *The International Conference on Theoretical and Applied Physics*, vol. 1011, May, 2018.
- [9] M. Guzairy, N. Mohamad, and A. R. Yunus, “Enhancing Student’s Understanding in Entrepreneurship Through Business Plan Simulation”, *MATEC Web of Conferences* 150, 05091 (2018), Volume 150, Feb, 2018.